ASG-SmartTest™ TCA
User's Guide

Version 7.0
Publication Number: STT0200-70
Publication Date: February 2003

ASG Worldwide Headquarters Naples Florida USA | asg.com | info@asg.com
1333 Third Avenue South, Naples, Florida 34102 USA Tel: 941.435.2200 Fax: 941.263.3692 Toll Free: 800.932.5536 (USA only)

Contents

Preface

This *ASG-SmartTest TCA User's Guide* describes how to use ASG-SmartTest - Test Coverage Analysis (TCA), (herein called SmartTest-TCA). SmartTest-TCA is provided as an enhancement to ASG-SmartTest (herein called SmartTest) and adds functionality by allowing you to define test coverage for a group of programs, helping to ensure that all program routines have been executed in the testing process.

Allen Systems Group, Inc. (ASG) provides professional support to resolve any questions or concerns regarding the installation or use of any ASG product. Telephone technical support is available around the world, 24 hours a day, 7 days a week.

ASG welcomes your comments, as a preferred or prospective customer, on this publication or on any ASG product.

About this Publication

This publication consists of these chapters:

- Chapter 1, "Introduction," provides an overview of SmartTest-TCA.
- Chapter 2, "Using SmartTest-TCA: An Overview," provides a high-level overview of the TCA processes.
- Chapter 3, "Preparing SmartTest-TCA Tests," describes how to prepare to use and access TCA.
- Chapter 4, "Creating and Maintaining TCA Plans," describes how to create and maintain a TCA plan.
- Chapter 5, "Running TCA Test Sessions," describes how to select a TCA plan and begin a test session using online, batch, IMS/DC, or CICS environments.
- Chapter 6, "Recording Non-TCA Test Results into a TCA Plan," describes how to record test results from a non-TCA test session into a TCA plan.
- Chapter 7, "Setting Breakpoints with Impact Datasets," describes how to create or import impact datasets.
- Chapter 8, "Using TCA Session Profiles," describes how to specify and save a TCA profile.

- Chapter 9, "Maintaining Coverage Results in the Test Utility Screen," describes how to view and manipulate test coverage results.
- Chapter 10, "Generating Reports," describes how to generate reports after testing programs and save the test result data.
- Chapter 11, "TCA Commands, Scripts, and Profiles," provides information about the TCA commands, and on recording commands and profiles into a script file.

Related Publications

The documentation library for ASG-SmartTest consists of these publications (where *nn* represents the product version number):

- *ASG-Center Installation Guide* (CNX0300-*nn*) contains installation and maintenance information for ASG-Center, the common set of libraries shared by the ASG-ESW suite of products.
- *ASG-ESW Enhancement Summary* (ESW1000-*nn*) highlights the new functionality for this release.
- *ASG-SmartTest CICS User's Guide* (STC0200-*nn*) contains specific commands and test session setup information for the CICS environments.
- *ASG-SmartTest for COBOL and Assembler User's Guide* (STA0200-*nn*) contains introductory and usage information for COBOL and Assembler. It also contains test session setup information for the TSO, ISPF, IMS/DB, DB/2, BTS, and Batch environments.
- *ASG-SmartTest IMS User's Guide* (STM0200-*nn*) contains specific commands and test session setup information for the IMS/DC environments.
- *ASG-SmartTest Installation Guide* (STX0300-*nn*) contains information for installing and maintaining ASG-SmartTest.
- *ASG-SmartTest PLI User's Guide* (STL0200-*nn*) contains introductory and usage information about how to use ASG-SmartTest with the PL/I language. It also contains test session setup information for the TSO, ISPF, IMS/DB, DB/2, BTS, and Batch environments.

- *ASG-SmartTest Quick Start for COBOL/ASM* (STA0900-*nn*) summarizes how to use ASG-SmartTest with the COBOL or Assembler language.
- *ASG-SmartTest Quick Start for PL/I* (STL0900-*nn*) summarizes how to use ASG-SmartTest with the PL/I language.
- *ASG-SmartTest Reference Guide* (STX0400-*nn*) contains detailed reference information about CUA pull-downs and pop-ups, ASG-SmartTest command syntax, and pseudo code.
- *ASG-SmartTest Reference Summary* (STX0600-*nn*) summarizes the syntax and usage of ASG-SmartTest commands.
- *ASG-SmartTest TCA User's Guide* (STT0200-*nn*) contains procedures for using the ASG-SmartTest-TCA (Test Coverage Analysis) option.

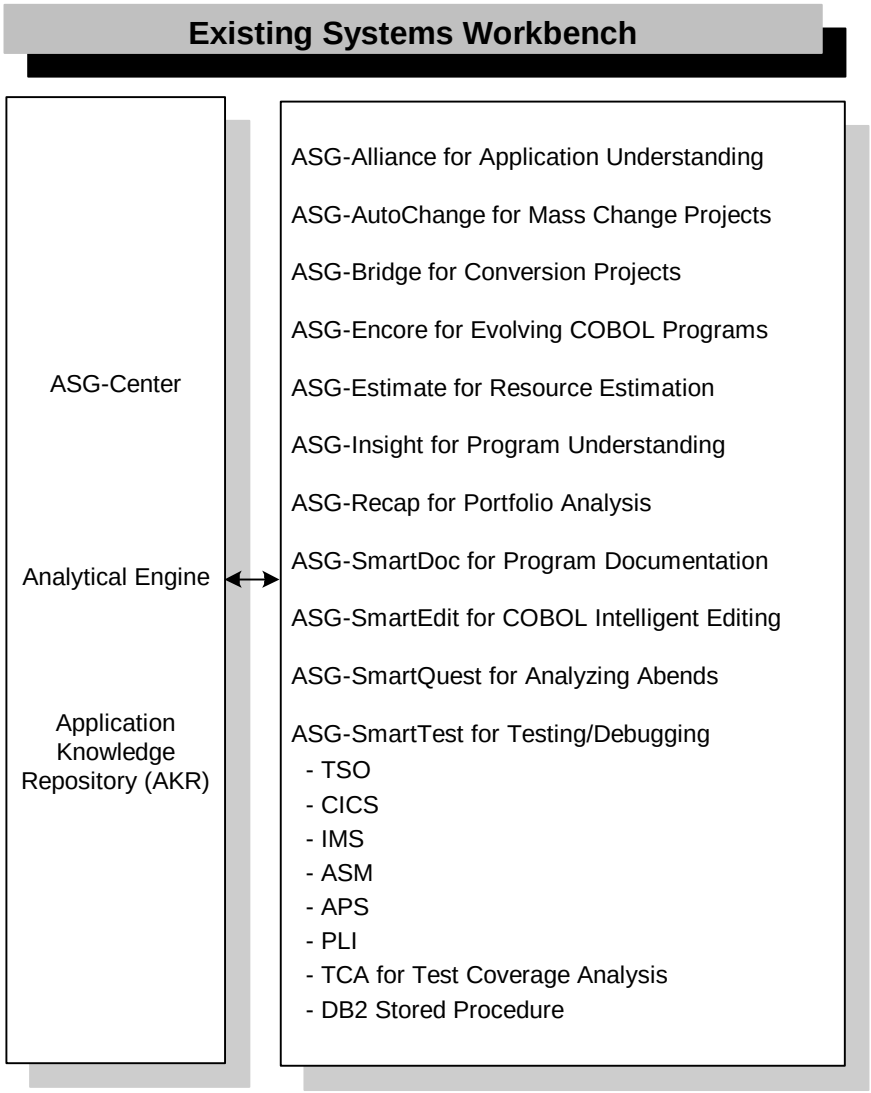
Note:

To obtain a specific version of a publication, contact ASG Customer Support.

ASG-Existing Systems Workbench (ASG-ESW)

ASG-ESW (herein called ESW) is an integrated suite of components designed to assist organizations in enhancing, redeveloping, or re-engineering their existing systems. ESW products use the Application Knowledge Repository (AKR) to store source program analysis information generated by the Analytical Engine. Figure 1 represents the components of ESW.

Figure 1 • ASG Existing Systems Workbench



This table contains the name and description of each ESW component:

ESW Product	Herein Called	Description
ASG-Alliance	Alliance	The application understanding component that is used by IT professionals to conduct an analysis of every application in their environment. Alliance supports the analysis and assessment of the impact of change requests upon an entire application. Alliance allows the programmer/ analyst to accurately perform application analysis tasks in a fraction of the time it would take to perform these tasks without an automated analysis tool. The impact analysis from Alliance provides application management with additional information for use in determining the resources required for application changes.
ASG-AutoChange	AutoChange	The COBOL code change tool that makes conversion teams more productive by enabling quick and safe changes to be made to large quantities of code. AutoChange is an interactive tool that guides the user through the process of making source code changes.
ASG-Bridge	Bridge	The bridging product that enables field expansion for program source code, without being required to simultaneously expand the fields in files or databases. Because programs are converted in smaller groups, or on a one-by-one basis, and do not require file conversion, testing during the conversion process is simpler and more thorough.
ASG-Center	Center	The common platform for all ESW products. Center provides the common Analytical Engine to analyze the source program and store this information in the AKR. This common platform provides a homogeneous environment for all ESW products to work synergistically.

ESW Product	Herein Called	Description
ASG-Encore	Encore	The program re-engineering component for COBOL programs. Encore includes analysis facilities and allows you to extract code based on the most frequently used re-engineering criteria. The code generation facilities allow you to use the results of the extract to generate a standalone program, a callable module, a complement module, and a CICS server. Prior to code generation, you can view and modify the extracted Logic Segment using the COBOL editor.
ASG-Estimate	Estimate	The resource estimation tool that enables the user to define the scope, determine the impact, and estimate the cost of code conversion for COBOL, Assembler, and PL/I programs. Estimate locates selected data items across an application and determines how they are used (moves, arithmetic operations, and compares). Time and cost factors are applied to these counts, generating cost and personnel resource estimates.
ASG-Insight	Insight	The program understanding component for COBOL programs. Insight allows programmers to expose program structure, identify data flow, find program anomalies, and trace logic paths. It also has automated procedures to assist in debugging program abends, changing a computation, and resolving incorrect program output values.
ASG-Recap	Recap	The portfolio analysis component that evaluates COBOL applications. Recap reports provide function point analysis and metrics information, program quality assessments, intra-application and inter-application comparisons and summaries, and historical reporting of function point and metrics information. The portfolio analysis information can also be viewed interactively or exported to a database, spreadsheet, or graphics package.
ASG-SmartDoc	SmartDoc	The program documentation component for COBOL programs. SmartDoc reports contain control and data flow information, an annotated source listing, structure charts, program summary reports, exception reports for program anomalies, and software metrics.

ESW Product	Herein Called	Description
ASG-SmartEdit	SmartEdit	The COBOL editing component that can be activated automatically when the ISPF/PDF Editor is invoked. SmartEdit provides comprehensive searching, inline copybook display, and syntax checking. SmartEdit allows you to include an additional preprocessor (for example, the APS generator) during syntax checking. SmartEdit supports all versions of IBM COBOL, CICS, SQL, and CA-IDMS.
ASG-SmartQuest	SmartQuest	The diagnostic tool for analyzing batch and CICS transaction abends. SmartQuest has been designed to make the maximum use of simple point-and-shoot techniques to enable fast and easy navigation through any data dump.
ASG-SmartTest	SmartTest	The testing/debugging component for COBOL, PL/I, Assembler, and APS programs in the TSO, MVS Batch, CICS (including file services), and IMS environments. SmartTest features include program analysis commands, execution control, intelligent breakpoints, test coverage, pseudo code with COBOL source update, batch connect, disassembled object code support, and full screen memory display.

Invoking ESW Products

The method you use to invoke an ESW product depends on your system setup. If you need assistance to activate a product, see your systems administrator. If your site starts a product directly, use the ISPF selection or CLIST as indicated by your systems administrator. If your site uses the ESW screen to start a product, initiate the ESW screen using the ISPF selection or CLIST as indicated by your systems administrator and then typing in the product command on the command line.

The product names can also vary depending on whether you access a product directly or through ESW. See "ESW Product Integration" on page xii for more information about using ESW.

To initialize ESW products from the main ESW screen, select the appropriate option on the action bar pull-downs or type the product shortcut on the command line.

Product Name (ESW Name)	Shortcut	ESW Pull-down Options
Alliance (Application Understanding)	AL	Understand ▸ Application
AutoChange (Conversion Set)	CC	Change ▸ Conversion Set
Bridge	BR	Change ▸ ASG-Bridge
Encore (Program Re-engineering)	EN	Re-engineer ▸ Program
Estimate	ES	Measure ▸ ASG-Estimate
Insight (Program Understanding)	IN	Understand ▸ Program
Recap (Portfolio Analysis)	RC	Measure ▸ Portfolio
SmartDoc (Program Documentation)	DC	Document ▸ Program
SmartEdit	SE	Change ▸ Program **Or** Change ▸ Program with Options
SmartQuest	SQV	Understand ▸ Abend/Dump
SmartTest (Testing/Debugging)	ST	Test ▸ Module/Transaction

ESW Product Integration

Because ESW is an integrated suite of products, you are able to access individual ESW products directly, or through the main ESW screen. As a result, different fields, values, action bar options, and pull-down options display on a screen or pop-up depending on how you accessed the screen or pop-up.

Certain ESW products also contain functionality that interfaces with other ESW products. Using SmartTest as an example, if Alliance is installed, SmartTest provides a dynamic link to Alliance that can be used to display program analysis information. If Insight is installed and specified during the analyze, the Insight program analysis functions are automatically available for viewing logic/data relationships and execution path. For example, the Scratchpad option is available on the Options pull-down if you have Insight installed.

Access to these integrated products requires only that they be installed and executed in the same libraries.

Example 1

Figure 2 shows the Encore Primary screen that displays when you access Encore directly.

The Encore Primary screen contains these eight action bar menu items: File, View, Extract, Generate, Search, List, Options, and Help.

Figure 2 • Encore Primary Screen

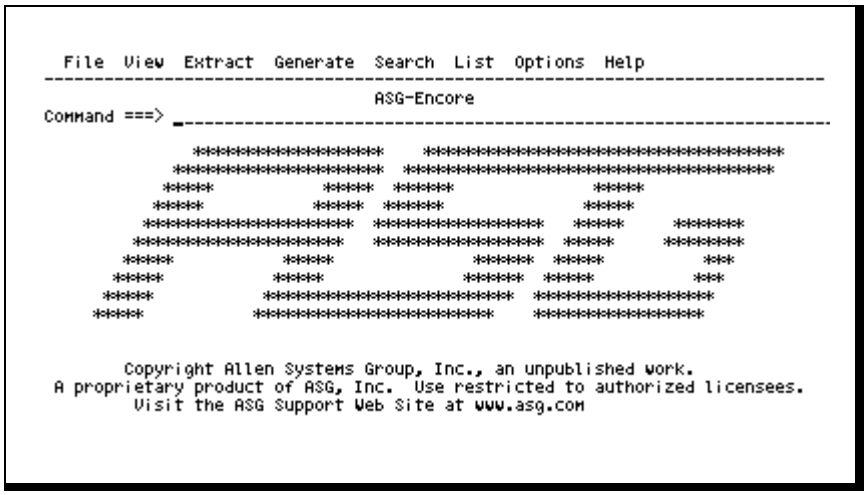
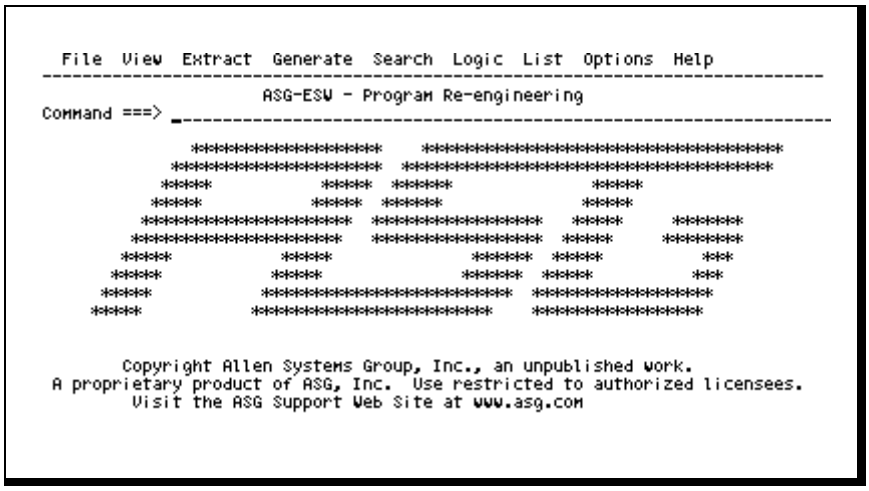


Figure 3 shows the Encore Primary screen that displays when you access Encore through ESW by selecting Re-engineer ▸ Program from the ESW action bar menu. Notice that the Primary screen name changes to ASG-ESW - Program Re-engineering when you enter Encore through ESW. Also, the Logic menu item displays if Insight is installed.

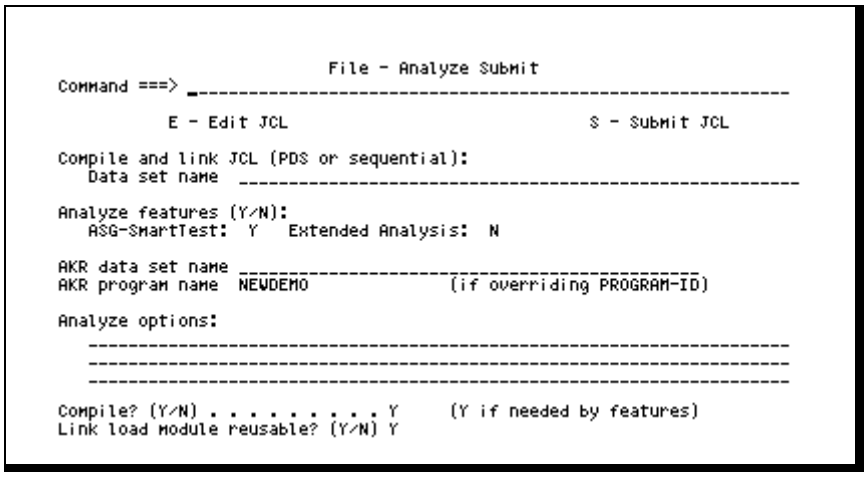
Figure 3 • ESW Encore Primary Screen



Example 2

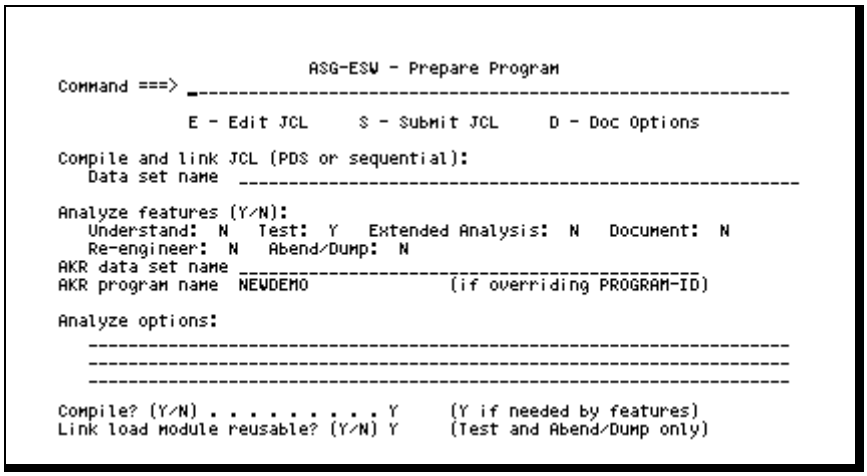
Figure 4 shows the File - Analyze Submit pop-up that displays when you access SmartTest directly. Figure 5 on page xv shows the File - Analyze Submit pop-up that displays when you access SmartTest through ESW.

Figure 4 • File - Analyze Submit Screen



The actions shown on these screens can also vary. For example, the D - Doc Options action is only available on the File Prepare Program screen (or File - Analyze Submit screen) if SmartDoc is installed on your system. In Figure 4 on page xiv, the Doc Options action is not displayed.

Figure 5 • ASG-ESW - Prepare Program Screen (accessed through ESW)



Notice that the Analyze features field in Figure 5 lists additional ESW products than shown on Figure 4 on page xiv. This field is automatically customized to contain the ESW products you have installed on your system. These are the names of the analyze types:

Analyze Type	Analyze Type (ESW)
ASG-Encore	Re-engineer
ASG-Insight	Understand
ASG-SmartDoc	Document
ASG-SmartQuest	Abend/Dump
ASG-SmartTest	Test
Extended Analysis (ASG-SmartTest with Insight installed)	Extended Analysis

Publication Conventions

ASG uses these conventions in technical publications:

Convention	Represents
ALL CAPITALS	Directory, path, file, dataset, member, database, program, command, and parameter names.
Initial Capitals on Each Word	Window, field, field group, check box, button, panel (or screen), option names, and names of keys. A plus sign (+) is inserted for key combinations (e.g., Alt+Tab).
lowercase italic monospace	Information that you provide according to your particular situation. For example, you would replace *filename* with the actual name of the file.
Monospace	Characters you must type exactly as they are shown. Code, JCL, file listings, or command/statement syntax.
	Also used for denoting brief examples in a paragraph.
Vertical Separator Bar (|) with underline	Options available with the default value underlined (e.g., Y|N).
Underline	Denotes a cursor-selectable field or line.

ASG Customer Support

ASG provides support throughout the world to resolve questions or problems regarding installation, operation, or use of our products. We provide all levels of support during normal business hours and emergency support during non-business hours.

ASG Third-party Support. ASG provides software products that run in a number of third-party vendor environments. Support for all non-ASG products is the responsibility of the respective vendor. In the event a vendor discontinues support for a hardware and/or software product, ASG cannot be held responsible for problems arising from the use of that unsupported version.

Intelligent Support Portal (ISP)

Online product support is available at: http://www.asg.com/support/support.asp via the ASG Intelligent Support Portal (ISP). Your logon information for ISP online support is:

Customer ID = *NNNNNNNNN*
Password = *XXXXXXXXXX*

where:

NNNNNNNNN is your customer ID supplied by ASG Product Distribution.
XXXXXXXXXX is your unique password supplied by ASG Product Distribution.

The *ASG-Intelligent Support Portal User's Guide* provides instructions on how to use the ISP and is located on the ASG Support web page.

Telephone Support

To expedite response time, please have this information ready:

- Product name, version number, and release number
- List of any fixes currently applied
- Any alphanumeric error codes or messages written precisely as displayed
- A description of the specific steps that immediately preceded the problem
- Verify whether you received an ASG Service Pack or cumulative service tape for this product. It may include information to help you resolve questions regarding installation of this ASG product. The Service Pack instructions are in a text file on the distribution media included with the Service Pack. You can access the latest software corrections and Service Packs via the ISP.
- The severity code (ASG Customer Support uses an escalated severity system to prioritize service to our clients. The severity codes and their meanings are listed below.)

Severity Codes and Expected Support Response Times

Severity	Meaning	Expected Support Response Time
1	Production down, critical situation	Within 30 minutes
2	Major component of product disabled	Within 2 hours
3	Problem with the product, but customer has work-around solution	Within 4 hours
4	"How-to" questions and enhancement requests	Within 4 hours

The Americas

	Phone	Fax	E-mail
United States and Canada	800.354.3578	1.703.464.4901	support@asg.com

Europe, Middle East, and Africa (EMEA)

During normal business hours, we recommend that you call the Central Support number first (except in South Africa).

	Phone	Fax	E-mail
Central Support	00.800.3544.3578	44.1727.812018	support.emea@asg.com
English	44.1727.736305	44.1727.812018	support.uk@asg.com
French	33.141.028590	33.141.028589	support.fr@asg.com
German	49.89.45716.200	49.89.45716.400	support.de@asg.com
Italian	39.0290450025		support.it@asg.com
Dutch	31.30.241.6133		support.nl@asg.com
Spanish	34.913.523.800	34.917.156.961	support.es@asg.com
South Africa	800.201.423		support.sa@asg.com

Asia Pacific (APAC)

	Phone	Fax	E-mail
Central Support	61.3.9645.8500	61.3.9645.8077	support.au@asg.com
Australia	800.637.947	61.3.9645.8077	support.au@asg.com
Hong Kong	800.96.2800		support.hk@asg.com
Japan	81.3.5326.3684	81.3.5326.3001	support.au@asg.com
Singapore	65.224.3080	65.224.8516	support.sg@asg.com

All Other Countries (Also for any non-working numbers)

	Phone	Fax	E-mail
All other countries	1.239.435.2201		support@asg.com

If you receive a voice mail message, follow the instructions to report a production-down or critical problem. Leave a detailed message including your name and phone number. An ASG Customer Support representative will be paged and will return your call as soon as possible. Please have available the information described previously when the ASG Customer Support representative contacts you.

ASG Documentation/Product Enhancements

Submit all product and documentation suggestions to ASG's product management team at http://www.asg.com/asp/emailproductsuggestions.asp.

If you do not have access to the web, FAX your suggestions to product management at (239) 263-3692. Please include your name, company, work phone, e-mail ID, and the name of the ASG product you are using. For documentation suggestions include the publication number located on the publication's front cover.

1 Introduction

This chapter provides an overview of SmartTest-TCA, and contains these sections:

An Overview of SmartTest-TCA

SmartTest-TCA is a tool for program, application, and system testing. It can be used during development and quality assurance phases and is provided as an add-on to SmartTest. SmartTest-TCA enhances SmartTest functionality by allowing you to define test coverage for a group of programs, which helps to ensure that all program routines have been executed in the testing process.

Features

Each time you execute a program within the SmartTest-TCA environment, SmartTest-TCA records and maintains information about the test session. SmartTest-TCA enables you to reliably repeat coverage tests and record the results by resetting the test environment to that used in previous test sessions.

SmartTest-TCA records results at the individual statement level and reports results at the statement, paragraph, and program levels.

Benefits

SmartTest-TCA provides these benefits:

- Allows you to confirm actual program execution
- Allows you to partition test cases to focus testing on modified programs
- Produces reports showing where a program did or did not execute
- Allows you to identify test cases that duplicate coverage

Language, Environment, and Database Support

SmartTest-TCA runs under MVS (ESA) TSO/ISPF (Release 3.5 through 4.8) and supports these languages, execution environments, and databases.

Languages

- CA Optimizer II (COBOL II)
- COBOL II Release 3 and 4
- CASE-generated COBOL
- COBOL/370
- COBOL for MVS and VM
- Enterprise COBOL Release 3.1
- High-level Assembler
- INTERSOLV APS
- OS PL/I Versions 2.3
- PL/I MVS & VM

Execution Environments

- CICS 4.1
- CICS/TS 1.1, 1.2, 1.3, and 2.1
- HOGAN
- IMS/DB
- IMS/DC
- ISPF Dialog Manager
- LANGUAGE ENVIRONMENT
- TSO

Databases

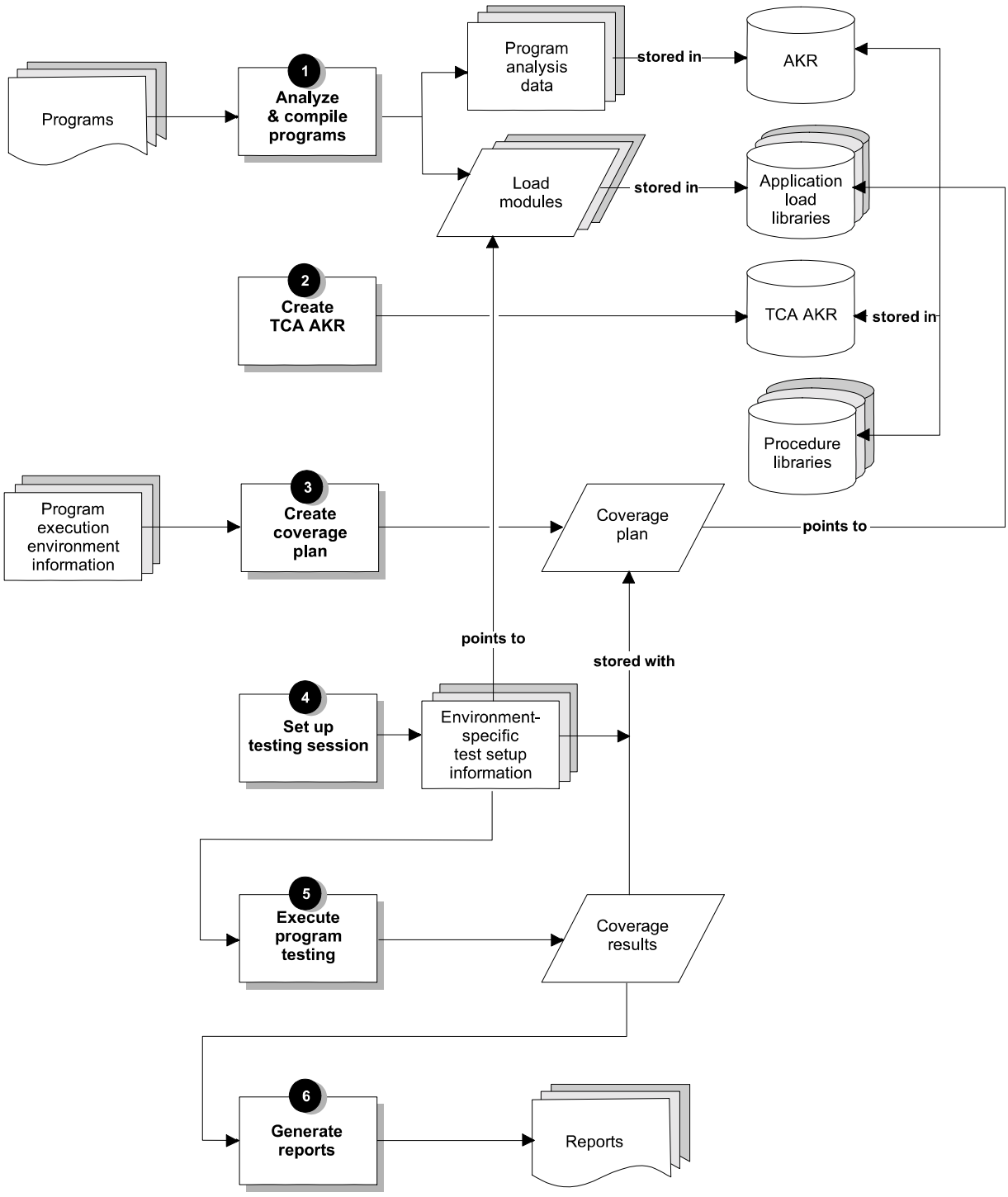
- DATACOM/DB
- DB2
- IDMS/DB
- IMS
- SYSTEM 2000
- TOTAL/TIS
- VSAM

2 Using SmartTest-TCA: An Overview

This chapter describes the steps necessary to use SmartTest-TCA, and contains these sections:

Figure 6 provides a conceptual overview of the TCA process.

Figure 6 • TCA Process Overview



Step 1 - Analyzing and Compiling the Programs to Test

You must analyze and compile the programs before testing. The ESW Analytical Engine stores program analysis data in the AKR. Load modules are stored in a load library.

See the *ASG-SmartTest for COBOL and Assembler User's Guide* or the *ASG-SmartTest Reference Guide* for more information about analyzing and compiling programs.

Step 2 - Creating a TCA AKR

Using a TCA-specific AKR is recommended to accommodate the large amount of data TCA generates.

See the *ASG-Center Installation Guide,* or the SmartTest user guide for your environment for more information about allocating an AKR.

Step 3 - Setting Up the TCA Test Session

Prepare the SmartTest-TCA test session by setting the parameters applicable to the environment for the program you will test. Note that the environment settings point to the load module and procedure libraries required to run the program during the test session. The environment settings are stored with the TCA Plan.

See "Preparing SmartTest-TCA Tests" on page 11 for more information.

Step 4 - Creating a TCA Plan

Create a TCA Plan specifying an execution environment and the programs you want to test. TCA Plans group programs for testing and saving test results. You can repeat tests by recalling test session settings from the TCA Plan. TCA Plans contain a default execution environment setting, but can store test results produced from different environments.

A TCA Plan contains pointers to these items:

- Compiled and analyzed programs belonging to the Plan
- The procedure libraries and load libraries used by the compiled and analyzed programs
- The AKR containing the program analyses

SmartTest-TCA stores TCA Plans in a TCA-specific AKR. A TCA Plan can use an unlimited number of programs, and up to eight load libraries and three AKRs.

See "Creating and Maintaining TCA Plans" on page 15 for more information.

Step 5 - Running the Test

When you begin a TCA test, you will enter a test session description and select a test result type (the type of data to gather during testing). Available test result types are:

- COVERAGE
- POINT

Note:

See the table on page 31 for more information about the COVERAGE and POINT test result types.

Depending on your execution environment, interaction during the test session may not be required. This table lists SmartTest-TCA test session behavior for different types of execution environments.

Execution Environment	Description
Batch	• No user interaction • No connection • Can execute multiple steps in a job • The TSO region is free to perform other tasks
CICS and IMS/DC	• Records after each command or transaction input • Completes on disconnect
TSO Online	• No user interaction

SmartTest-TCA records information about the executed program instructions and stores the recorded information as a coverage result for the TCA Plan. SmartTest-TCA creates a coverage result for each test session. Tests that do not complete successfully contain an abend message in place of their description.

Coverage results contain information about all programs executed during the test session. For example, a batch test session that executes three programs generates a single test coverage result.

You can initiate a new test session using the execution environment and program settings from a previous test session by selecting the coverage result from that test session. SmartTest-TCA generates a new coverage result after re-establishing the execution environment and program settings in which the original coverage result was generated.

After testing, you can export coverage results to a sequential file.

See "Running TCA Test Sessions" on page 23 for more information.

Step 6 - Generating Reports

You can generate reports using one or more coverage results in a TCA Plan. This table describes the reports available in SmartTest-TCA.

Report	Description
Detail Count and Program Summary	Lists the number of times each program instruction executed.
Detail Execution	Specifies whether each program instruction executed.
Program/ Paragraph Label Count	Lists the number of times each paragraph executed.
Program/ Paragraph Label Execution	Specifies whether each paragraph executed.
Executed/Not Executed Summary	Lists number of statements executed or not executed.

Note:

See "Generating Reports" on page 53 for more information about generating SmartTest-TCA reports.

3 Preparing SmartTest-TCA Tests

This chapter describes how to prepare to run SmartTest-TCA tests, and contains these sections:

Setting up TCA Tests

To set up a TCA test

1 Run a SmartTest compile/analyze on the programs you want to include in your TCA Plan.

See the SmartTest user guide for your environment for more information about compiling and analyzing programs.

2 Allocate an AKR specifically for TCA.

SmartTest-TCA generates a large amount of information for each TCA Plan and each program tested. Using a TCA-specific AKR ensures adequate space for storing TCA-related data.

This is a list of SmartTest-TCA's estimated AKR usage by element.

Element	Size
TCA Plan	4096 bytes/record
Coverage result	4096 bytes plus 4096 bytes per 500 PROCEDURE DIVISION statements

See the *ASG-Center Installation Guide* or your SmartTest user's guide for more information about allocating AKRs.

3 Test and debug each program.

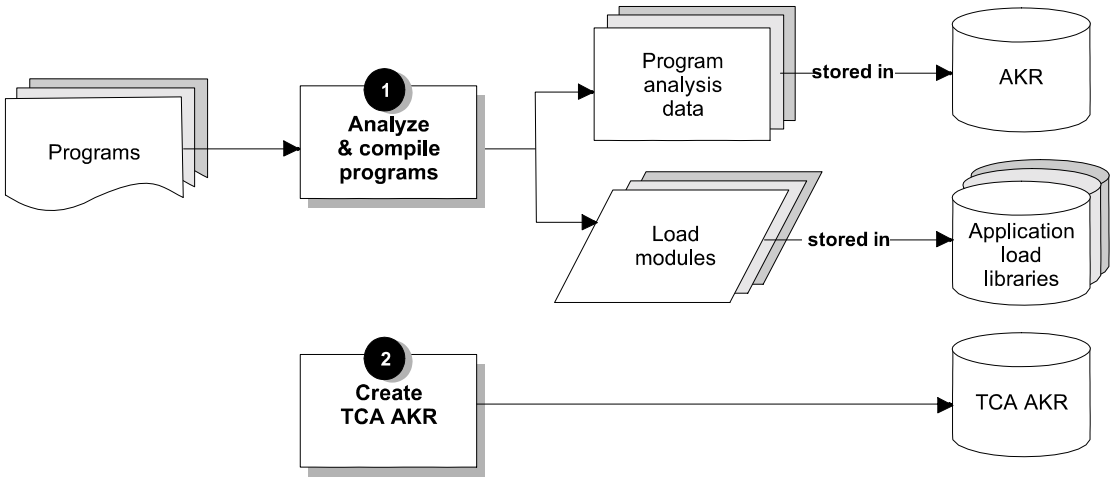
4 Ensure the programs you plan to test execute successfully. If necessary, test your programs first using SmartTest to debug them.

Note:

See the *ASG-SmartTest for COBOL and Assembler User's Guide* or the *ASG-SmartTest PL/I User's Guide* for information about running non-TCA tests.

Figure 7 shows the process of preparing to test with TCA.

Figure 7 • Preparing to Test with TCA



Starting SmartTest-TCA

To start SmartTest-TCA

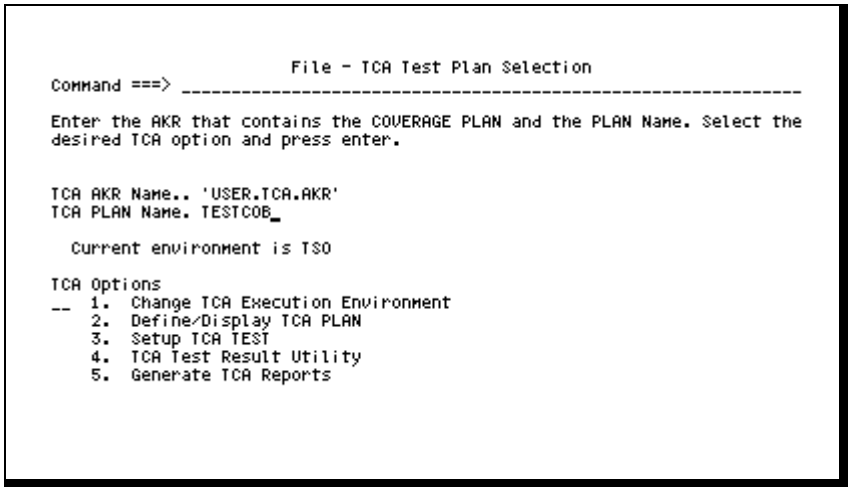
1 Select File ▸ Test Coverage from the SmartTest primary screen.

Or

Type `LIST TCA` on the command line and press Enter.

The File - TCA Test Plan Selection screen, shown in Figure 8, displays.

Figure 8 • File - TCA Test Plan Selection Screen



After you have a TCA Plan defined, you can use LIST TCA to open the Plan by typing this command:

`LIST TCA` *`testplanname`*

2 Type the name of the TCA AKR and the name of the test coverage plan.

3 Use the TCA Options to organize programs for testing and specify your environment settings. See "Creating and Maintaining TCA Plans" on page 15 for more information about these options.

4 Creating and Maintaining TCA Plans

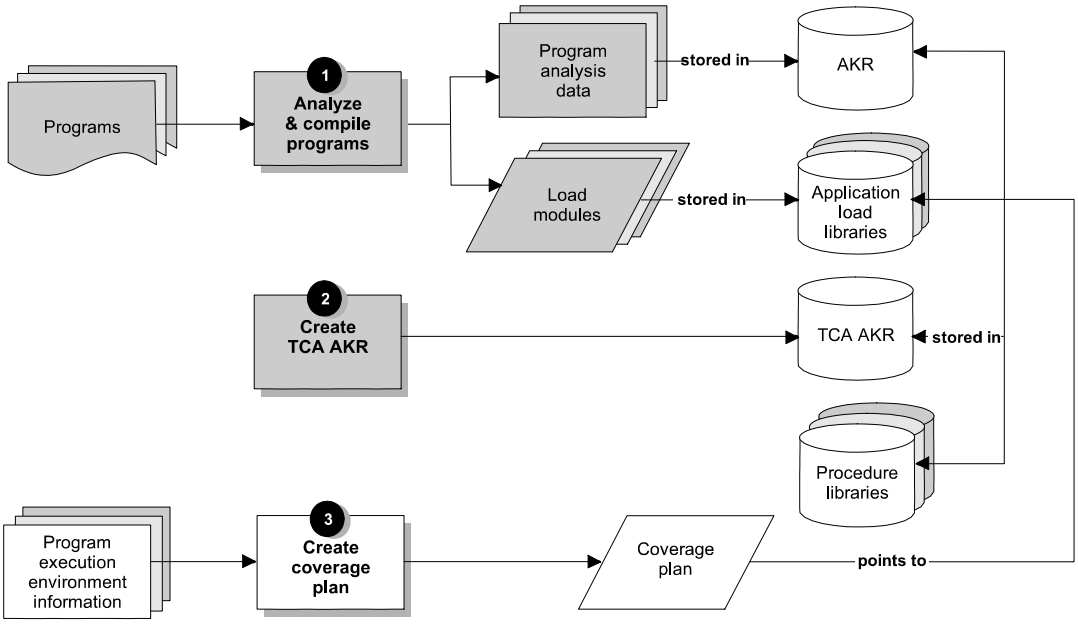
This chapter describes how to create and use SmartTest-TCA plans, and contains these sections:

Using a TCA Plan

SmartTest-TCA uses TCA Plans to organize programs for testing and environment settings in which testing occurs. You must create a TCA Plan before you can test programs with SmartTest-TCA. The TCA Plan is stored in the TCA-specific AKR. When you select an existing TCA Plan, SmartTest-TCA sets your execution environment to the one defined in the plan.

Figure 9 shows how creating a TCA Plan fits into the process of preparing for TCA testing.

Figure 9 • Creating a TCA Plan



Creating a TCA Plan

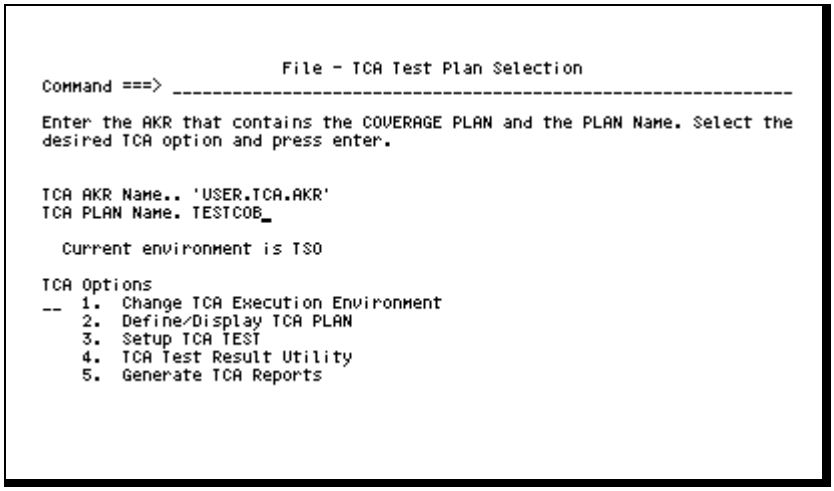
Before creating a TCA Plan, complete the preliminary steps listed in ""Preparing SmartTest-TCA Tests" on page 11. After you have entered the SmartTest-TCA facilities, as described in "Starting SmartTest-TCA" on page 13, you can create a TCA Plan.

Step 1 - Naming the TCA Plan

To name the TCA Plan

1 Select File ▸ Select Test Coverage, or type `LIST TCA`, and press Enter. The File - TCA Test Plan Selection pop-up, shown in Figure 10, displays.

Figure 10 • File - TCA Test Plan Selection Pop-up

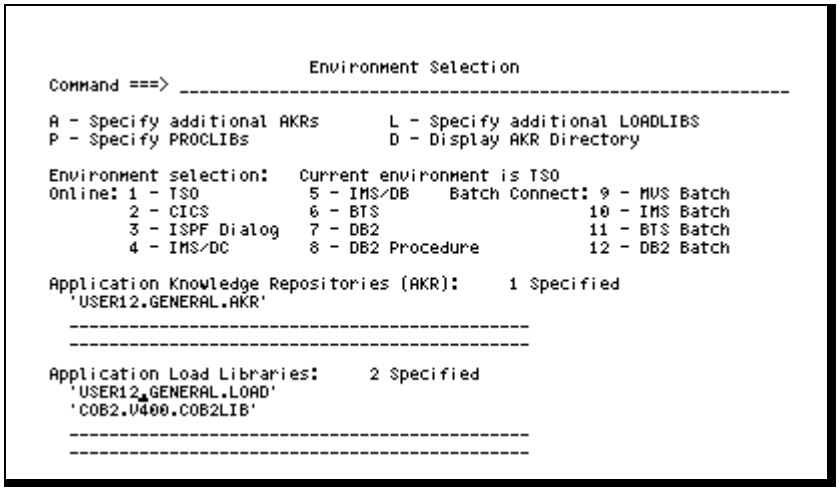


2 Type the name of the TCA AKR in which you want to save the TCA Plan.

3 Enter a TCA Plan name. This name can be a maximum of eight alphanumeric characters.

4 From the TCA Options list, select Change TCA Execution Environment and press Enter. The Environment Selection pop-up, shown in Figure 11, displays.

Figure 11 • Environment Selection Pop-up



This table describes the TCA Options available from the File - TCA Test Plan Selection pop-up.

Option	Description
Change TCA Execution Environment	Activates the Environment Selection screen so you can set or change the TCA Plan's execution environment. (See Figure 11 on page 17.)
Define/Display TCA Plan	Activates the TCA Coverage Plan screen so that you can define/display the TCA Plan. (See Figure 12 on page 20.)
Setup TCA Test	Activates a setup screen for the environment specified so that you can setup a TCA test session. (See Figure 17 on page 26 for an example of a session setup screen.)
TCA Test Result Utility	Activates the T.C.A Test Utility screen so that you can list the coverage results stored in the TCA Plan. (See Figure 28 on page 50.)
Generate TCA Reports	Activates the T.C.A. Report Selection screen so that you can generate TCA reports. (See Figure 29 on page 55.)

Step 2 - Defining the Default Execution Environment

When you create a TCA Plan, it inherits the default environment settings from the most recent test session you executed.

To assign different default environment settings to the TCA Plan

1. Select the Change TCA Execution Environment option on the File - TCA Test Plan Selection pop-up to display the Environment Selection pop-up.

2. Edit the entries in the Application Knowledge Repositories and Application Load Libraries fields as necessary.

   - The Application Knowledge Repositories field should contain the name of the AKR where the analyzed programs you want to test reside.
   - The Application Load Libraries field should contain the name of the load library that holds the load modules created during the compile/analyze process for the programs you want to include in the TCA Plan.

3 Select an environment and proceed with the environment settings as you would in a non-TCA test. When the environment settings are in place, you are prompted to run a test session. Press PF3 until you return to the File - TCA Test Plan Selection pop-up.

Note:

After you have created a TCA Plan, you cannot change its AKRs or load libraries. If you attempt to change them and then open the TCA Plan, SmartTest-TCA ignores the changes and restores the original TCA Plan settings.

See the *ASG-SmartTest Reference Guide* and the *ASG-SmartTest for COBOL and Assembler User's Guide* for detailed information about execution environments.

Step 3 - Specifying the Programs to Test

Use the T.C.A. - Coverage Plan screen to perform these functions:

- Specify the TCA-specific AKR.
- Name the TCA Plan.
- Add or modify the description for the plan.
- Add or modify the designated TCA Profile PDS.
- Add or delete programs included in the TCA Plan.

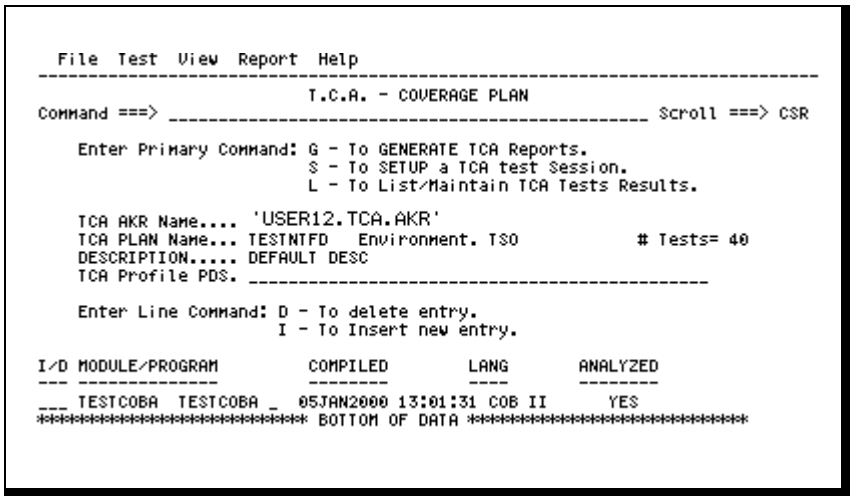
Note:

See "Using TCA Session Profiles" on page 45" for more information about the TCA profile PDS.

To add a program to a TCA Plan

1 Select the Define/Display TCA Plan option from the TCA Options field on the File - TCA Test Plan Selection pop-up and press Enter. The T.C.A. - Coverage Plan screen, shown in Figure 12, displays.

Figure 12 • T.C.A. - Coverage Plan



2 Type `I` in the I/D field and press Enter to open a blank line.

3 Enter module and program names. When adding modules or programs, you can use an asterisk as a wildcard character to add all programs in the AKR, or all programs starting with a substring. For example, TEST* would match all modules or programs whose names begin with TEST.

4 Press Enter to accept the modifications. A message confirming the plan update displays.

Note:

Make sure that a confirmation message displays after any change. If there is no confirmation message, the change was not applied.

For each program, TCA displays this information:

- Last compilation date
- The language in which it was written
- Whether it was analyzed for SmartTest

SmartTest-TCA checks that the program in the AKR and the load module match by comparing their compile dates and times. When the AKR program and load module compile dates and times do not match, SmartTest-TCA displays the message `...TIME STAMPS VARY....` To correct the error, reanalyze and relink the program.

To delete a program from a TCA Plan

1. Move the cursor to the line of the program to delete.
2. Type `D` in the I/D field and press Enter. The line for the program is deleted.

Note:

Program insertions are only reflected in test results and reports generated after the insertion. Reports generated with test results created before the insertion show the new programs as undefined.

Other TCA Plan-related Activities

You can enter explicit TCA commands on the command line on the T.C.A. - Coverage Plan screen to perform additional TCA activities. These commands are listed in this table:

Command	Description
G	Generates reports (see "Generating Reports" on page 53).
S	Sets up a test session (see "Running TCA Test Sessions" on page 23).
L	Lists results (see "Maintaining Coverage Results in the Test Utility Screen" on page 49).

Viewing All TCA Plans in an AKR

To view all TCA Plans in an AKR

1 Select File ▸ Select Test Coverage, or type `LIST TCA,` and press Enter. The File - TCA Test Plan Selection pop-up displays.

2 Type an asterisk (*) in the TCA PLAN Name field and press Enter. The TCA - Active Plans screen, shown in Figure 13, displays.

Figure 13 • T.C.A. - Active Plans Screen

```
                          T.C.A. - ACTIVE PLANS
Command ===> _____________________________________________ Scroll ===> CSR

    TCA AKR Name.. 'USER.TCA.AKR'

    Enter Line Commands: D - To Delete Test Plan
                         S - To Select Test Plan for review

D/S  TCA PLAN      DESCRIPTION                              TESTS
---  ---------     ---------------------------------------  -------
****************************** TOP OF DATA ********************************
__                 DEFAULT DESC                                 1
__   DEADCODE      DEFAULT DESC                                 3
__   DEADPLAN      DEFAULT DESC                                 1
__   DEADTEST                                                   5
__   DEADTES2      TESTING                                      4
__   DE19068       DEFAULT DESC                                 5
__   TESTCICS      DEFAULT DESC                                 2
__   TESTCOB       DEFAULT DESC                                32
__   TESTCOBA      DEFAULT DESC                                13
__   TESTCOBC      DEFAULT DESC                                 2
__   TESTH         DEFAULT DESC                                 0
__   TESTMRG       VIAMERGE TEST                                8
```

3 Select a TCA Plan for review or deletion.

Note:

If you delete a TCA Plan, you delete all its associated coverage results.

5 Running TCA Test Sessions

This chapter describes how to run SmartTest-TCA test sessions, and contains these sections:

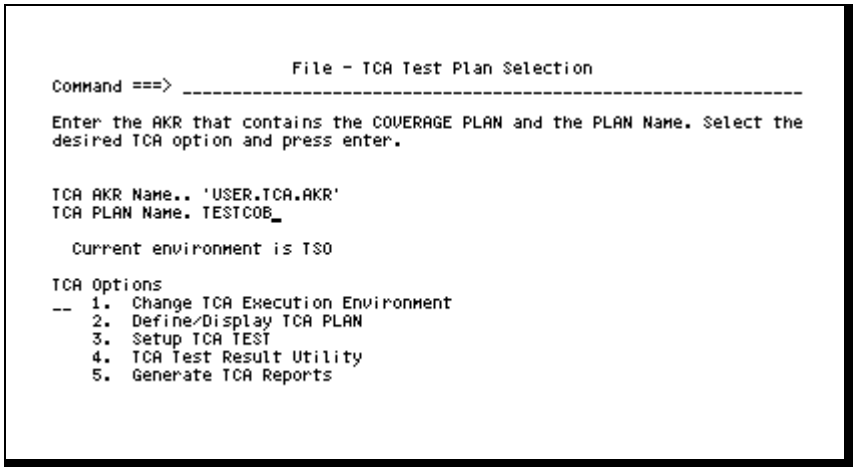
Each time you run a TCA test session, TCA records test results containing information about the program instructions executed. Test coverage results are stored in the TCA-specific AKR along with the TCA Plan.

Step 1 - Selecting a TCA Plan

To select a TCA Plan

1 Select File ▸ Select Test Coverage. The File - TCA Test Plan Selection screen, shown in Figure 14, displays.

Figure 14 • File - TCA Test Plan Selection Screen

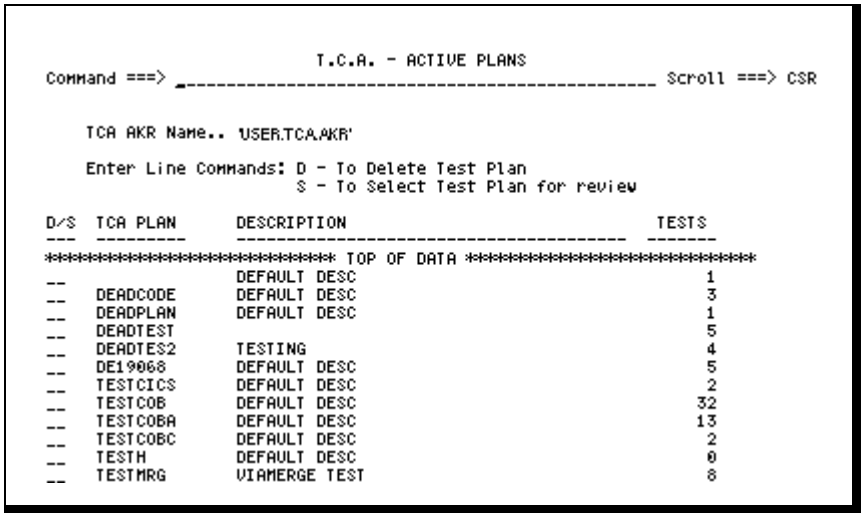


2 Type the name of the TCA AKR and the TCA Plan. If you do not know the name of your TCA Plan, see the next section.

To select a TCA Plan from the T.C.A. - Active Plans pop-up

1 If you do not know the name of the TCA Plan, leave the TCA PLAN Name field on the File - TCA Test Plan Selection screen blank and press Enter. The T.C.A. - Active Plans pop-up, shown in Figure 15, displays, which lists all TCA Plans available in the AKR.

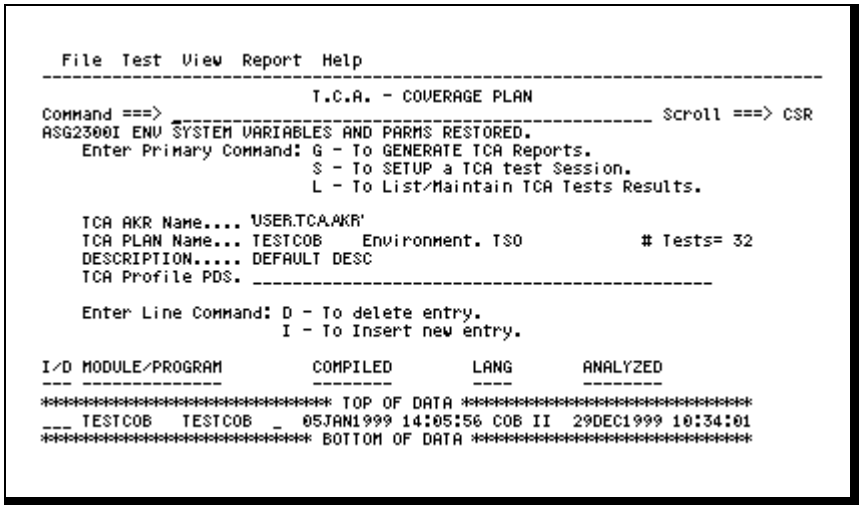
Figure 15 • T.C.A. - Active Plans Pop-up



```
                          T.C.A. - ACTIVE PLANS
Command ===> _____________________________________________ Scroll ===> CSR

    TCA AKR Name.. 'USER.TCA.AKR'

    Enter Line Commands: D - To Delete Test Plan
                         S - To Select Test Plan for review

D/S  TCA PLAN      DESCRIPTION                              TESTS
---  ---------     ---------------------------------------  -------
****************************** TOP OF DATA ***************************
__                 DEFAULT DESC                               1
__   DEADCODE      DEFAULT DESC                               3
__   DEADPLAN      DEFAULT DESC                               1
__   DEADTEST                                                 5
__   DEADTES2      TESTING                                    4
__   DE19068       DEFAULT DESC                               5
__   TESTCICS      DEFAULT DESC                               2
__   TESTCOB       DEFAULT DESC                              32
__   TESTCOBA      DEFAULT DESC                              13
__   TESTCOBC      DEFAULT DESC                               2
__   TESTH         DEFAULT DESC                               0
__   TESTMRG       VIAMERGE TEST                              8
```

2 Type S to the left of the TCA Plan you want to select and press Enter. The T.C.A. - Coverage Plan screen, shown in Figure 16, displays.

Figure 16 • T.C.A. - Coverage Plan Pop-up



```
  File  Test  View  Report  Help
--------------------------------------------------------------------------------
                          T.C.A. - COVERAGE PLAN
Command ===> _____________________________________________ Scroll ===> CSR
ASG2300I ENV SYSTEM VARIABLES AND PARMS RESTORED.
    Enter Primary Command: G - To GENERATE TCA Reports.
                           S - To SETUP a TCA test Session.
                           L - To List/Maintain TCA Tests Results.

    TCA AKR Name.... 'USER.TCA.AKR'
    TCA PLAN Name... TESTCOB    Environment. TSO           # Tests= 32
    DESCRIPTION..... DEFAULT DESC
    TCA Profile PDS. _____________________________________________

    Enter Line Command: D - To delete entry.
                        I - To Insert new entry.

I/D MODULE/PROGRAM         COMPILED        LANG       ANALYZED
--- --------------         --------        ----       --------
****************************** TOP OF DATA ***************************
___ TESTCOB   TESTCOB   _  05JAN1999 14:05:56 COB II  29DEC1999 10:34:01
****************************** BOTTOM OF DATA ************************
```

Step 2 - Starting the TCA Test Session

To start a TCA test session

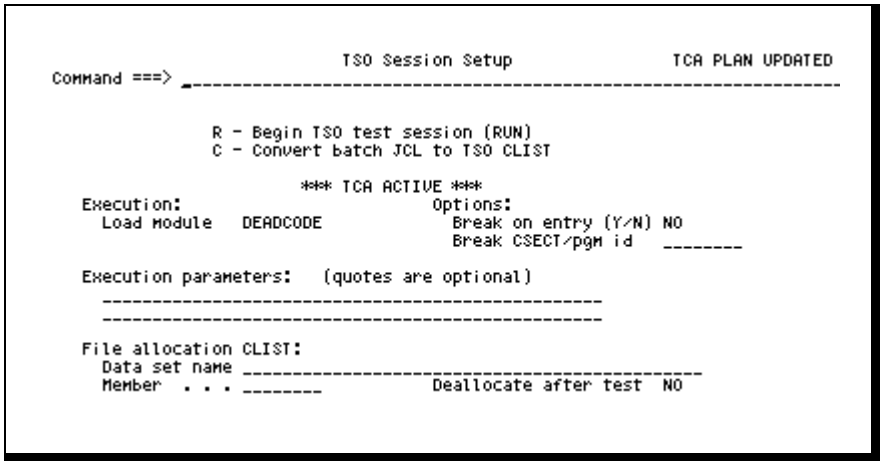
1 Select the Define/Display TCA PLAN option on the File - TCA Test Plan Selection pop-up and press Enter. The T.C.A. - Coverage Plan screen displays.

Or

If you selected the TCA Plan from the TCA - Active Plans pop-up, type S on the command line and press Enter.

2 Type S on the command line and press Enter. An environment-specific session setup screen, similar to Figure 17, displays. The session setup screen you see is specific to the execution environment you designated as the default when you created the TCA Plan. Notice the TCA PLAN UPDATED message.

Figure 17 • TSO Session Setup



3 Type R on the command line and press Enter to begin the test session.

You can cancel the TCA session and proceed with a non-TCA session using the current execution environment setup by typing SETUP NOTCA and pressing Enter. Because no user interaction occurs during TCA test runs (other than user interaction required by the execution environment), modifying the Break on Entry and Break CSECT/pgm id fields does not affect TCA testing.

Step 3 - Selecting a Program and a Test Execution Environment

When entering a session setup screen the TCA Plan's default settings display. If you want to test a different program or test in a different environment, you need to make changes.

Changing the Program to Test

To test a program other than the one named in the Load module field on the session setup screen, enter the new program name in the Load module field. You can only select from those programs that are defined in the TCA Plan. To test a program that is not currently part of the plan, you must first add it to the TCA Plan.

Note:

See "Step 3 - Specifying the Programs to Test" on page 19 for instructions about adding programs to a TCA Plan.

Changing the Execution Environment

TCA Plans are not limited to a single execution environment. You can run a test, or a group of tests, in several different execution environments and save the test results to a single TCA Plan.

Note:

If you change the execution environment by selecting a non-TCA test profile using the LIST PROFILES command, you automatically set your test environment to non-TCA.

To change the execution environment for a single TCA test

1. Select a TCA Plan and proceed to the session setup screen for the default execution environment defined in the TCA Plan.
2. Type `ENV` on the command line and press Enter. The Environment Selection pop-up displays.
3. Select a different execution environment. A session setup screen for the execution environment you selected displays. The AKR and load library settings cannot be changed.
4. Enter appropriate values in the session setup screen and run the test.

Note:

See the SmartTest online help, the *ASG-SmartTest COBOL and Assembler User's Guide*, or the *ASG-SmartTest PL/I User's Guide* for information about the various environment-specific session setup screens.

Testing in batch environments does not require a CLIST; however, testing in TSO sessions does. For test sessions requiring a CLIST, if you have not previously generated a CLIST from the program's execution JCL you need to do so now.

To generate the CLIST, follow this step:

- Type `C` on the command line of the TSO Session Setup screen and press Enter.

Batch Environment Settings

To edit the execution JCL in batch environments

1. Type `ENV` and press Enter to display the Environment Selection screen.
2. Select a batch environment option and press Enter. The Batch Session Setup screen displays.
3. Type `E` on the command line to edit the JCL.
4. For utility or driver programs that invoke a program in the test plan, enter this qualifying comment on the line immediately before the first execute statement that invokes the program:

   ```
   //* EXECUTION FOR {program name}
   ```

   Begin the comment in column 5 and the program name in column 19.

Note:

See the *ASG-SmartTest for COBOL and Assembler User's Guide* for detailed information about editing execution JCL.

IMS/DC Environment Settings

In IMS/DC, select the transactions that will be monitored during the test session.

To test in an IMS/DC environment

1. Type `ENV` and press Enter to display the Environment Selection screen.
2. Select the IMS/DC environment option and press Enter. The IMS/DC Session Setup screen displays.
3. Type `T` on the command line and press Enter. The IMS/DC Select Transactions pop-up displays.
4. Select List all eligible transactions and press Enter. The IMS/DC Transaction Definition pop-up displays.

5 Select the desired transactions, adding parameters if necessary, and press Enter.

6 Press PF3 twice to return to the session setup screen.

7 Type `C` on the command line and press Enter to connect to IMS.

CICS Environment Settings

In a CICS environment, you must define the transactions or programs to be monitored.

To test in a CICS environment

1 Type `ENV` and press Enter to display the Environment Selection screen.

2 Select the CICS environment option and press Enter. The CICS Session Setup screen displays.

3 Select Request Monitoring (Protection Tables) and press Enter. The User Protection Menu displays.

4 Select Task or Program and press Enter. The Task Specification or Program Specification screen displays.

5 Specify the transaction ID or the program to be monitored.

6 Press PF3 twice to return to the CICS Session Setup screen.

7 Type `C` on the command line and press Enter to connect to CICS.

Note:

In the TCA Plan, you must include the program that issues the EXEC CICS RETURN command.

Step 4 - Selecting a Test Result Type

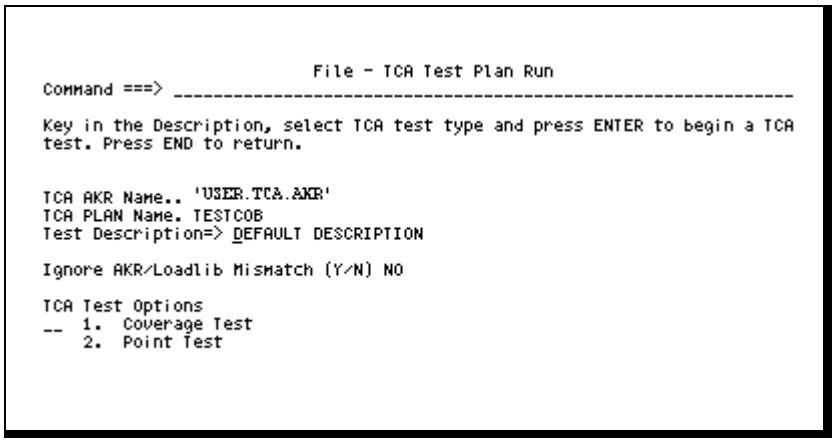
To select a test result type

1 Type `LI TCA` on the command line and press Enter. The File - TCA Test Plan Selection screen displays.

2 Select Setup TCA TEST under TCA Options. The session setup screen redisplays with the message *** TCA ACTIVE ***.

3 Enter the required commands to begin testing in the current environment. For example, if you have a TSO environment, type R on the command line of the TSO - Session Setup screen and press Enter. The File - TCA Test Plan Run pop-up, shown in Figure 18, displays. The TCA AKR Name and TCA PLAN Name fields show the names specified in the TCA Plan.

Figure 18 • File - TCA Test Plan Run Pop-up



4 Type a test description in the Test Description field. The test description is stored as a comment in the test coverage result.

5 Specify whether you want SmartTest-TCA to ignore mismatched timestamps on the AKR and Loadlib. If you specify Yes in the Ignore AKR/Loadlib Mismatch field, TCA runs to completion even if a timestamp mismatch is detected. If you specify No, the TCA test ends when a timestamp mismatch is encountered.

Note:

If you type Yes in the Ignore AKR/Loadlib Mismatch field, TCA reporting results may be inaccurate and unreliable.

6 Under TCA Test Options, enter the number that corresponds to the type of test results you want to gather and press Enter to start the test session.

This table describes the available types of test coverage results.

Type	Data Gathered
Coverage	Records the number of times each executable line of code was executed.†
Point	Records whether each statement following a breakpoint was executed‡

† During TCA testing, all SmartTest options such as breakpoints, tailoring, pseudo code, and break on entry are disabled.

‡ To gather TCA point results, you must save breakpoints to the source code before TCA testing.

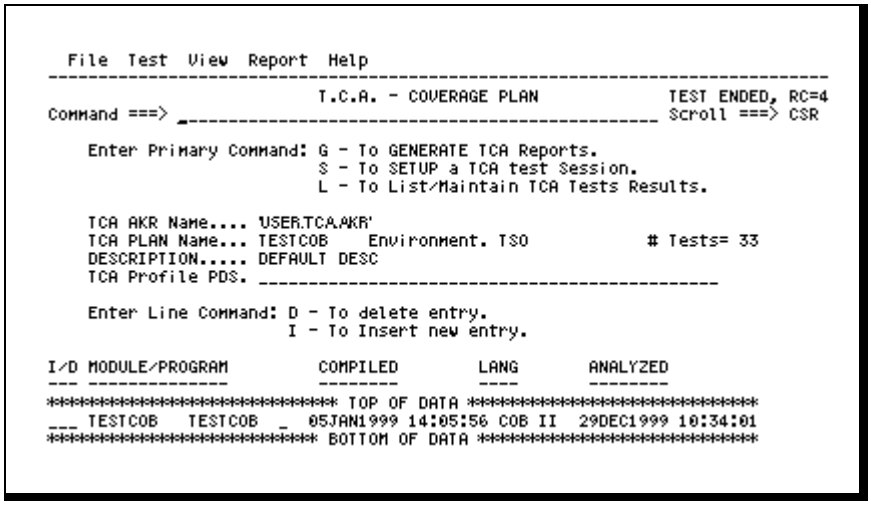
Testing in Online, Batch, IMS/DC, or CICS Environments

These sections describe how testing progresses from the session setup screen in online, batch, IMS/DC, and CICS environments.

Testing in Online Environments

In TSO, ISPF, IMS/DB, and DB2 environments, SmartTest-TCA executes the test session without user interaction. The terminal is not released until the test is complete. When the test is complete, SmartTest-TCA displays the session setup screen appropriate for your environment with a `TEST ENDED` message and a return code in the upper right corner, as shown in Figure 19.

Figure 19 • Example of a Testing Ended Screen



If the test ends unsuccessfully, a `TEST CANCELLED` message displays. When a test ends unsuccessfully, SmartTest-TCA stores the test session's error message and return code as the test result description.

Testing in Batch Environments

In MVS, DB2, and IMS/DB batch environments, the test runs without dialog between the TSO session and the batch job.

To test in MVS, DB2, and IMS/DB batch environments

1. After setting up a TCA test session, selecting a test option (COVERAGE or POINT), and running the test, the system displays a `JOB SUBMITTED` message. Press Enter to return to the File - TCA Test Plan Run screen. A `JOB SUBMITTED` message displays in the upper right corner of the screen.

2. Press PF3 to return to the session setup screen.

3. You can check the SmartTest queue to see the status of submitted jobs. On the session setup screen, type `C` on the command line and press Enter to display the Connect to Job screen.

4. Review the status of the submitted job. The Connect to Job screen shows one entry for each step in a TCA batch job. If the job abends or has JCL errors, remove each entry by typing `P` in the Select column for that entry and pressing Enter. When the job successfully completes, it is no longer displayed in the Connect to Job screen.

Testing in IMS/DC or CICS Environments

In IMS/DC (Native or 3270 emulation) and CICS environments, when the session starts, a `SmartTest Active` message displays.

Issue transactions and code data entry until testing completes. When testing finishes, return to SmartTest-TCA:

To disconnect in IMS, follow this step:

- Issue the /RCL command and press Enter. You are returned to the session setup screen.

To disconnect in CICS, follow this step:

- Use the toggle key to return to the session setup screen. Type `D` on the command line and press Enter.

6 Recording Non-TCA Test Results into a TCA Plan

This chapter describes how to record test results from a non-TCA test session into a TCA Plan, and contains these sections:

Recording non-TCA test results into a TCA Plan is useful for capturing development test session information running under programmer control, and for extending the percentage of code executed beyond what would be driven by data alone.

Before you begin, complete these steps:

- Ensure that the program you want to test is included in a TCA Plan. See "Step 3 - Specifying the Programs to Test" on page 19 for more information about adding a program to a TCA Plan.
- Set breakpoints, modify data, and use GO TO commands to extend the percentage of code executed, as desired.
- Record the non-TCA test results into a TCA Plan using the TCA RECORD command. The information recorded is the same as that displayed when issuing a LIST COUNTS command during an active test session.

Note:

See the *ASG-SmartTest Reference Guide*, the *ASG-SmartTest for COBOL and Assembler User's Guide*, the *ASG-SmartTest PL/I User's Guide*, or the online help for detailed information about the LIST COUNTS command.

Step 1 - Preparing and Running a Non-TCA Test Session

To run a non-TCA test session

1. Select an environment for the non-TCA test session using the Environment Selection screen.
2. Modify the test session using the LIST TAILOR command.
3. Complete the setup.
4. Begin the non-TCA test session.

Note:

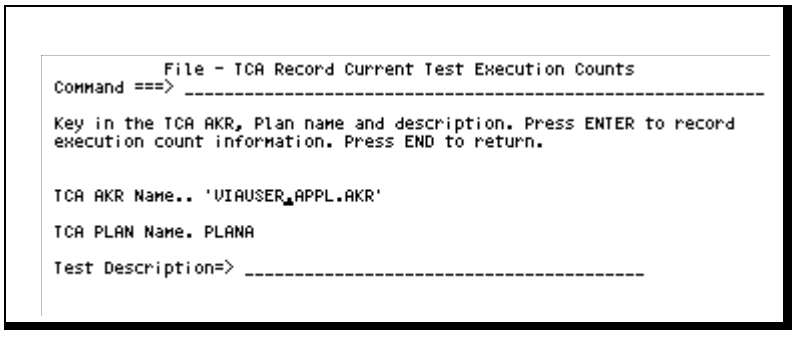
See the *ASG-SmartTest for COBOL and Assembler User's Guide*, the *ASG-SmartTest Reference Guide*, or the *ASG-SmartTest PL/I User's Guide* for more information about the LIST TAILOR command.

Step 2 - Executing a TCA RECORD

To execute a TCA RECORD

1. Type `TCA RECORD`. You can issue this command any time during the test session. If you do not specify all parameters on the command line, the File - TCA Record Current Test Execution Counts pop-up displays (see Figure 20).

Figure 20 • The File - TCA Record Current Test Execution Counts Pop-up



If you know all the parameter values, you can use this syntax to execute a TCA RECORD:

`TCA RECORD,AKR=`*`akrname`*`,PLAN=`*`planname`*`,DESC=`*`ddd`*

where:

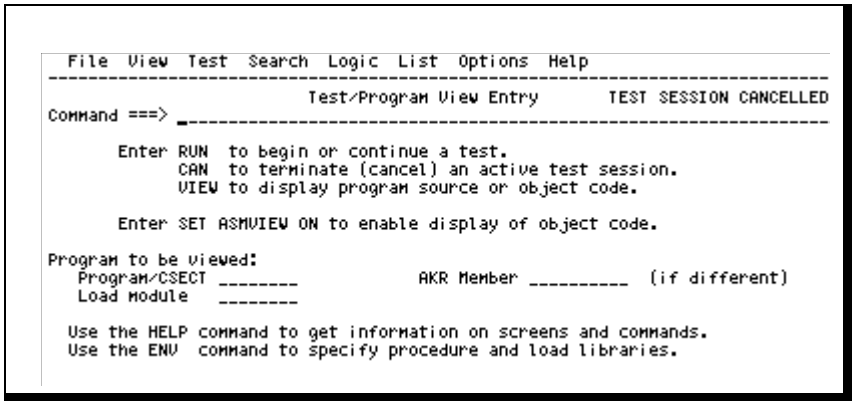
`akrname` is the name of the TCA AKR that contains the TCA Plan.

`planname` is the name of the TCA Plan that contains the program.

`ddd` is a description of the TCA Plan.

2 Complete the fields on the File - TCA Record Current Test Execution Counts pop-up and press Enter. The test is canceled and the Test/Program View Entry screen displays with the message `TEST SESSION CANCELLED` in the upper right corner, as shown in Figure 21.

Figure 21 • The Test/Program View Entry Screen



7 Setting Breakpoints with Impact Datasets

This chapter describes how to set breakpoints using impact datasets, and contains these sections:

SmartTest provides facilities for automatically setting breakpoints with lists of data items. The list is called an impact dataset. You can create the impact dataset yourself or import it from AutoChange, Alliance, or Estimate.

Using Impact Datasets From Other Programs

Execution JCL supplied with these ESW products is used to run batch jobs for creating impact datasets:

- AutoChange — VIAMEXPJ
- Alliance — VIABEXCV
- Estimate — VIAJEXCV

Note:

See the *ASG-AutoChange User's Guide*, the *ASG-Alliance User's Guide*, or the *ASG-Estimate User's Guide* for information about creating impact datasets.

Using User-supplied Impact Datasets

The name of an impact dataset must match the name of the program to which it will be applied. When creating an impact dataset, create it as a member of a PDS and name it appropriately.

The member contains a list of fully qualified datanames, one per line. Each dataname must begin in column ten. Lines beginning with an asterisk are comment lines. All lines that are not comment lines must contain an X in each of the first four columns.

Note:

Fully qualified datanames must begin in column ten.

This is an example of a user-supplied impact dataset:

```
*   THIS IS A COMMENT LINE
XXXX     TEST-DATE
XXXX     NUMBER1 OF TEST-NUM1
XXXX
XXXX
```

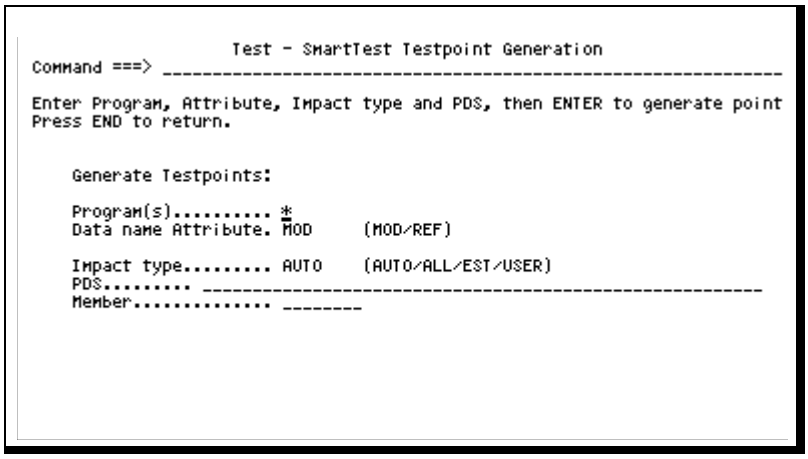
Setting Breakpoints with Impact Datasets

The steps used to set breakpoints with impact datasets are the same whether the impact dataset is produced by the user or by an external program.

To set breakpoints with impact datasets

1 Type `TESTPOINT` on any SmartTest-TCA screen command line and press Enter. The Test - SmartTest Testpoint Generation pop-up, shown in Figure 22, displays.

Figure 22 • The Test - SmartTest Testpoint Generation Pop-up



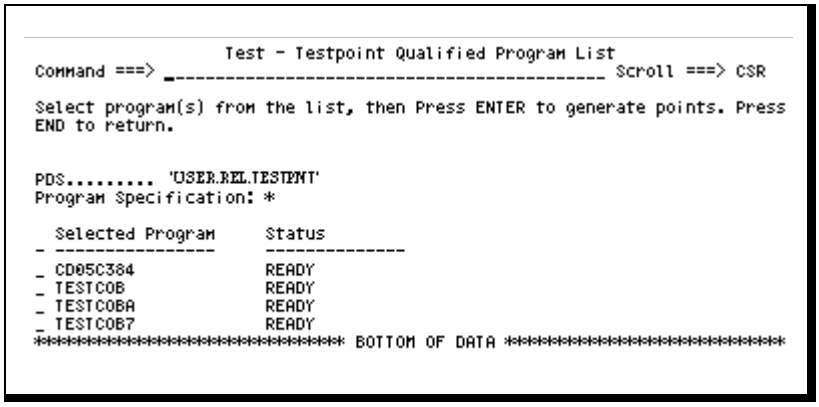
2 Edit the text fields entries in the Test - SmartTest Testpoint Generation pop-up using the information in this table:

Field	Description
Program(s)	Selects programs listed in the impact dataset. Enter the name of a single program or use an asterisk to select all of them. The names you select must match the names of the programs to which you apply the breakpoints.
Data Name Attribute	Specifies where to set breakpoints. One of two entries is allowed, MOD or REF. MOD sets a breakpoint wherever an impacted data item is modified. REF sets a breakpoint wherever the impacted data item is referenced.

Field	Description
Impact Type	Indicates the source of the imported impact dataset. Valid entries are: **AUTO.** AutoChange **ALL.** Alliance **EST.** Estimate **USER.** User-supplied
PDS	Specifies the PDS containing the impact dataset to be imported.
Member	For the impact types ALL, EST, or USER, lists the PDS member name containing the impact dataset.

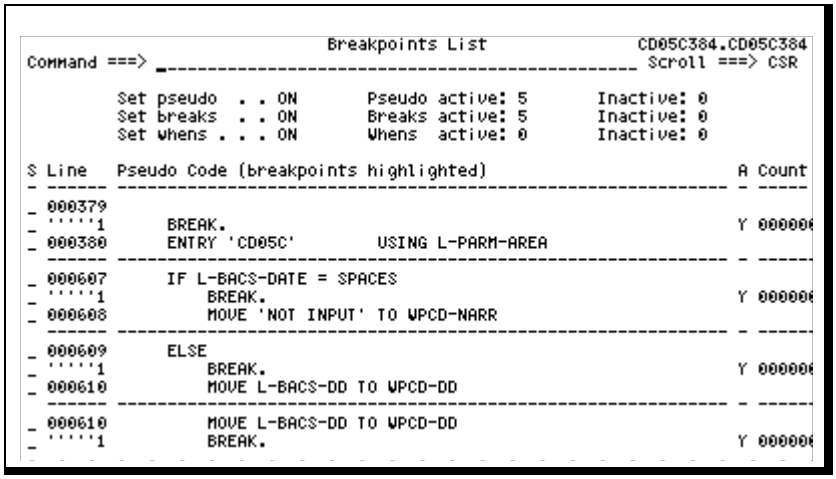
3 Press Enter to display the Test - Testpoint Qualified Program List pop-up, shown in Figure 23. This pop-up lists the programs you selected in the Test - SmartTest Testpoint Generation pop-up.

Figure 23 • The Test - Testpoint Qualified Program List



4 Type S next to the programs you want to select and press Enter. The Breakpoints List screen, shown in Figure 24, displays for the first program you selected.

Figure 24 • Breakpoints List Screen



5 Make any of these adjustments to the existing breakpoints:

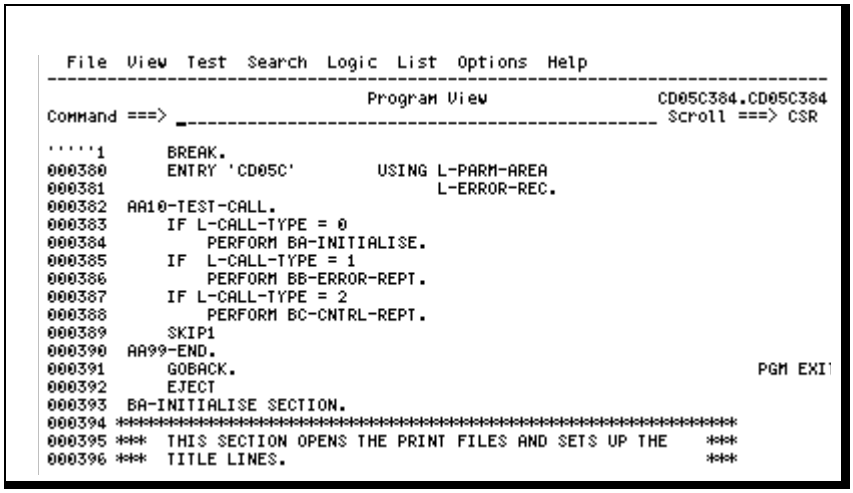
- Change the status of SET PSEUDO, SET BREAKS, or SET WHENS
- Deactivate individual breakpoints
- Remove individual breakpoints

Note:

See the *ASG-SmartTest for COBOL and Assembler User's Guide* for more information about the Breakpoints List screen.

6 When you are finished, exit the Breakpoints List screen. The Program View screen, shown in Figure 25, displays. The Program View screen displays breakpoints and the source code or disassembled object code and data.

Figure 25 • The Program View Screen

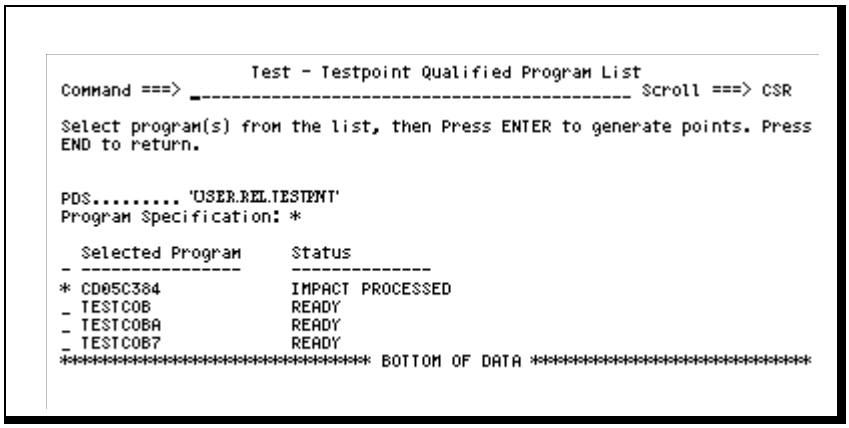


Note:

When running multiple point tests on the same source code, SmartTest-TCA assumes the points are the same in each test. If you change the points between tests, the results can be unpredictable when the report program attempts to match points between tests. If the number of breakpoints changes, SmartTest-TCA cannot identify the correspondence of points between tests and cannot reliably determine whether the data tested all the points.

7 Review the existing breakpoints and any changes you made. Press PF3 to exit the Program View screen and return to the Test - Testpoint Qualified Program List screen. If you successfully applied the impact dataset to a program, the `IMPACT PROCESSED` status is assigned to the program as shown in Figure 26.

Figure 26 • The Testpoint Qualified Program List (IMPACT PROCESSED)



Note:

The `IMPACT PROCESSED` status indicates that breakpoints have been saved into the program's source code in the AKR.

8 After applying an impact dataset to a program, you have these options:

- Press Enter and repeat step 5 on page 41 through step 7 for the next selected program on the Test - Testpoint Qualified Program List screen.
- Select a different program by typing `S` to the left of the program name and press Enter. Repeat step 5 on page 41 through step 7.
- Press PF3 to Exit.

8 Using TCA Session Profiles

This chapter describes how to use SmartTest-TCA profiles, and contains these sections:

The T.C.A. Coverage Plan screen allows you to specify a TCA profile. The TCA profile PDS can be the same one used for non-TCA test case profiles (usually your ISPF profile), or it can be another PDS allocated specifically for storing profiles.

Using a TCA profile allows you to save TCA settings and recall them by using the LIST PROFILE command. Selecting a TCA session profile activates the TCA Plan saved to that session profile. When setting up new TCA tests that are variations of previous TCA tests, you can recall TCA settings from a TCA profile PDS, make adjustments, and save the new settings back as a new TCA session profile.

TCA profiles can be any of these:

- Used for non-TCA test case profiles
- Used as your ISPF profile
- A PDS specifically allocated for storing TCA profiles

Note:

See *ASG-SmartTest for COBOL & Assembler User's Guide*, *ASG-SmartTest PL/I User's Guide*, or *ASG-SmartTest Reference Guide* for more information about the LIST PROFILES command.

Saving a TCA Session Profile

By default, non-TCA session profiles are saved to your ISPF profile. However, when you save TCA session profiles, you must specify the PDS in which the settings will be stored. You can change a TCA Plan's profile PDS at any time.

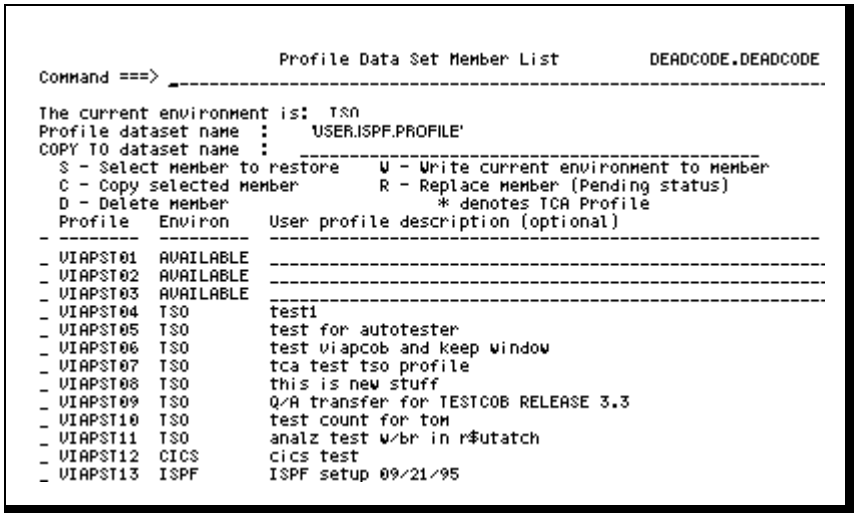
1 Type a TCA profile PDS in the TCA profile PDS field of the T.C.A. - Coverage Plan screen. You can mix TCA and non-TCA session profiles in the specified PDS.

Note:

It may be easiest to use your ISPF profile as the TCA profile PDS. However, if you plan to save many session profiles, you should allocate a separate PDS designated just for session profiles or TCA session profiles. Make sure to allocate the PDS before attempting to use its name.

2 Set up a TCA Test session using the steps described in "Preparing SmartTest-TCA Tests" on page 11). The session setting for the test is saved to the TCA session profile. In the session setup screen for your selected execution environment, type `LIST PROFILES` on the command line and press Enter. The Profile Data Set Member List pop-up, shown in Figure 27, displays.

Figure 27 • Profile Data Set Member List Pop-up



The column on the left of the Profile Data Set Member List pop-up shows member names containing test session profiles that have already been saved to the PDS listed in the Profile dataset name field. An asterisk next to a member name indicates it contains a TCA session profile.

3 Position the cursor on a line marked Available and type `W`. The cursor moves to the description field.

4 Type in a description and press Enter. The session profile is written to the TCA profile PDS and the member name is marked with an asterisk in the Profile Data Set Member List screen.

Recalling a TCA Session Profile

Except for selecting a profile member name marked with an asterisk, recalling a TCA session profile is done in the same way you would recall a non-TCA session profile.

To recall a TCA session profile

1 Type `LIST PROFILE` on any SmartTest-TCA screen and press Enter.

2 Select a profile member name marked with an asterisk and run the test from the session setup screen.

9 Maintaining Coverage Results in the Test Utility Screen

This chapter describes how to view and manipulate test coverage results, and contains these sections:

To open the TCA Plan and list the test coverage results

1. On the File - TCA Test Plan Selection pop-up (Figure 10 on page 17), type the TCA Plan name and select TCA Test Result Utility from the TCA Options field.

2. Type `L` on the command line of the T.C.A. - Coverage Plan screen.

   Or

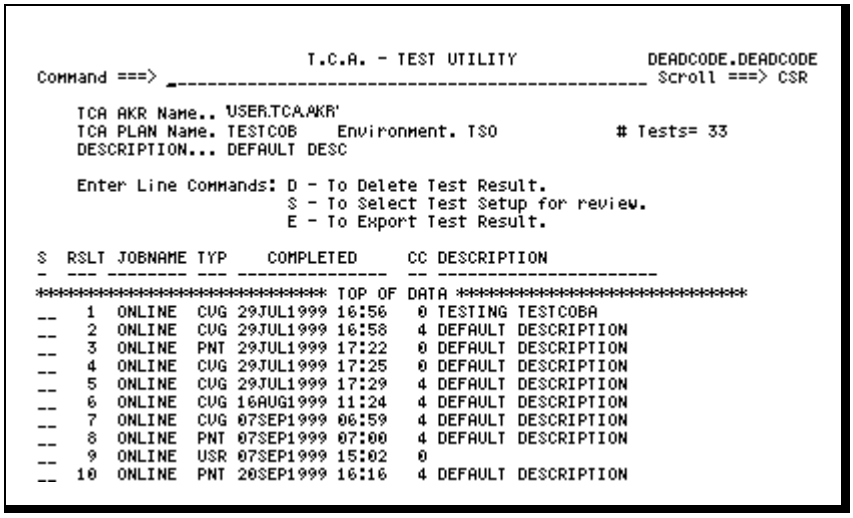
   From the command line of any SmartTest-TCA screen, type:

   `LIST TCA {`*`planname`*`} LIST`

   where *`planname`* is the name of the TCA Plan.

3 Press Enter to display the TCA - Test Utility screen, shown in Figure 28.

Figure 28 • The T.C.A. - Test Utility Screen



```
                          T.C.A. - TEST UTILITY              DEADCODE.DEADCODE
Command ===> _____________________________________________ Scroll ===> CSR

    TCA AKR Name.. 'USER.TCA.AKR'
    TCA PLAN Name. TESTCOB    Environment. TSO          # Tests= 33
    DESCRIPTION... DEFAULT DESC

    Enter Line Commands: D - To Delete Test Result.
                         S - To Select Test Setup for review.
                         E - To Export Test Result.

S  RSLT JOBNAME TYP    COMPLETED     CC DESCRIPTION
-  --- -------- --- ---------------  -- ----------------------
********************************* TOP OF DATA ********************************
__   1  ONLINE  CUG 29JUL1999 16:56   0 TESTING TESTCOBA
__   2  ONLINE  CUG 29JUL1999 16:58   4 DEFAULT DESCRIPTION
__   3  ONLINE  PNT 29JUL1999 17:22   0 DEFAULT DESCRIPTION
__   4  ONLINE  CUG 29JUL1999 17:25   0 DEFAULT DESCRIPTION
__   5  ONLINE  CUG 29JUL1999 17:29   4 DEFAULT DESCRIPTION
__   6  ONLINE  CUG 16AUG1999 11:24   4 DEFAULT DESCRIPTION
__   7  ONLINE  CUG 07SEP1999 06:59   4 DEFAULT DESCRIPTION
__   8  ONLINE  PNT 07SEP1999 07:00   4 DEFAULT DESCRIPTION
__   9  ONLINE  USR 07SEP1999 15:02   0
__  10  ONLINE  PNT 20SEP1999 16:16   4 DEFAULT DESCRIPTION
```

Line Commands in the TCA - Test Utility Screen

The S field in the result record supports these actions.

Command	Description
D	Deletes the test result.
S	Repeats the test session using the same program and environment settings used to produce this test coverage result. Typing S takes you to the appropriate session setup screen.
E	Exports the result. SmartTest-TCA exports its result to a VIAPUNCH file. Select the Log/List/Punch Definition menu item from the Options menu to begin the process of exporting a test result.

Note:

See the *ASG-SmartTest for COBOL and Assembler User's Guide* or the *ASG-SmartTest PL/I User's Guide* for detailed information about processing exported test coverage results.

Coverage Result Fields in the TCA - Test Utility Screen

This table describes the fields in each result record.

Field	Description
JOBNAME	Specifies the name of the job. These are the valid values: • A job name for batch testing • ONLINE – for online (TSO, IMS/DC, CICS) testing • NOT RUN – for results that could not be captured
TYP	Specifies the test result type. These are the valid values: • CVG – COVERAGE test • PNT – POINT test • USR - USER test **Note:** The USR test result type is valid for TCA RECORD only.
CC	Specifies the completion code. The return code accompanies a test session's `TEST ENDED` message. ABD indicates an ABEND.
DESCRIPTION	Provides a description of the test session. The description is the same text that was entered in the File - TCA Test Plan Run. If an abend occurred during testing, an abend description displays.

10 Generating Reports

This chapter describes how to generate SmartTest-TCA reports and contains these sections:

Selecting Result Data

You can generate reports after testing programs and saving the test result data. SmartTest-TCA provides reports using these types of test result data:

Data	Description
COVERAGE	Lists the number of times each executable line of code was executed.
POINT	Indicates whether each executable line of code following a break was executed.
USR (TCA RECORD only)	Specifies COVERAGE-type test data that resulted from a TCA RECORD.

By default, SmartTest-TCA uses all programs in a TCA Plan when generating reports. You can limit the programs shown in the report by selecting only specific programs.

This table describes the reports available in SmartTest-TCA.

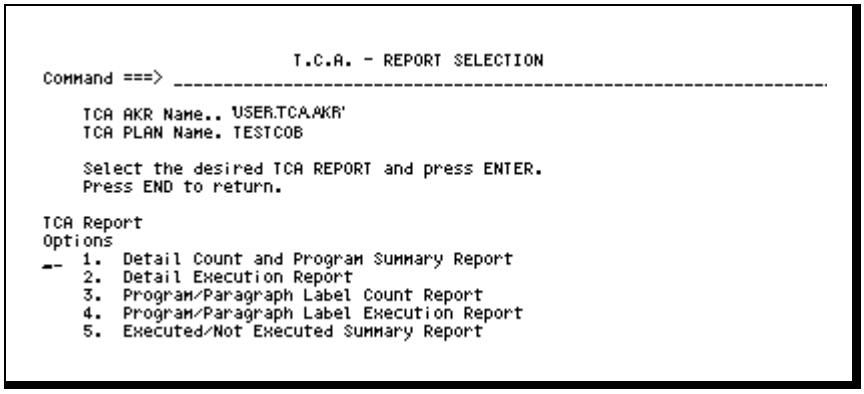
Report	Description
Detail Count and Program Summary	• Specifies the number of times each program instruction executed. • Reports at the program, paragraph/label, and statement level. • Compares the total number of executable statements to the total number of statements executed.
Detail Execution	• Indicates whether each program instruction executed. • Reports at the program, paragraph/label, and statement level. • Compares the total number of executable statements to the total number of statements executed.
Program/ Paragraph Label Count	• Specifies the number of times each paragraph executed. • Compares the total number of executable statements to the total number of statements executed.
Program/ Paragraph Label Execution	• Indicates whether each paragraph executed. • Compares the total number of executable statements to the total number of statements executed.
Executed/Not Executed Summary	• Specifies the number of statements executed or not executed. • Reports at the program level. • Shows the total count of statements executed/not executed and the percentage of statements executed/not executed as compared to the total number of statements in the program.

Generating Online Reports

To generate online reports

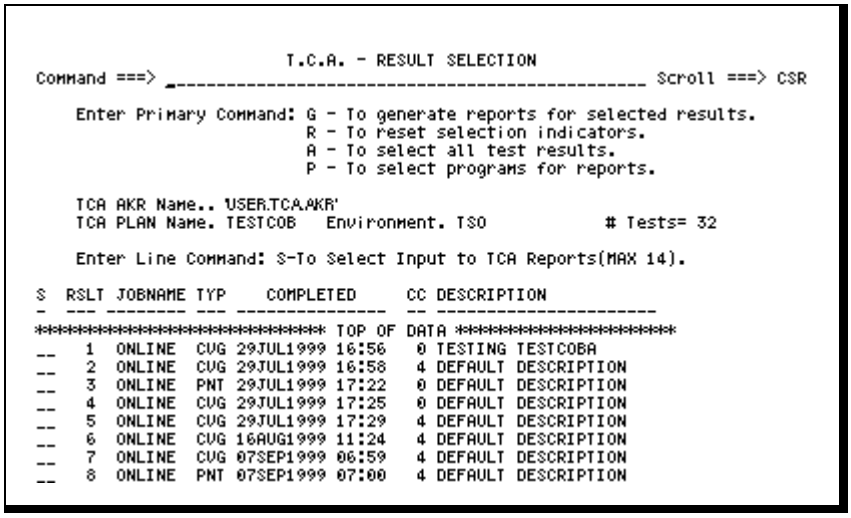
1 Type G on the command line in the T.C.A. - Coverage Plan screen and press Enter. The T.C.A. - Report Selection pop-up, shown in Figure 29, displays.

Figure 29 • T.C.A. - Report Selection Pop-up



2 Select a type of report from the TCA Report Options field and press Enter. The T.C.A. - Result Selection pop-up, shown in Figure 30, displays.

Figure 30 • Result Selection



3 Type S to select the results you want to produce a report for. You can specify a maximum of 14 results for a single report. If you select multiple results, they must be of the same type. For example, you cannot mix CVG and PNT results in a single report.

Selecting Results for a Report

To select a single result

1. Type S to the left of the result and press Enter.
2. Type G on the command line and press Enter.

To select multiple results

1. Type S to the left of each result you want to select and press Enter. You can select a maximum of 14 of the same type (CVG, PNT, or USR).
2. Type G on the command line and press Enter.

To select all results

1. Type A on the command line and press Enter.
2. Type G on the command line and press Enter.

To reset your result selection, follow this step:

- Type R on the command line and press Enter.

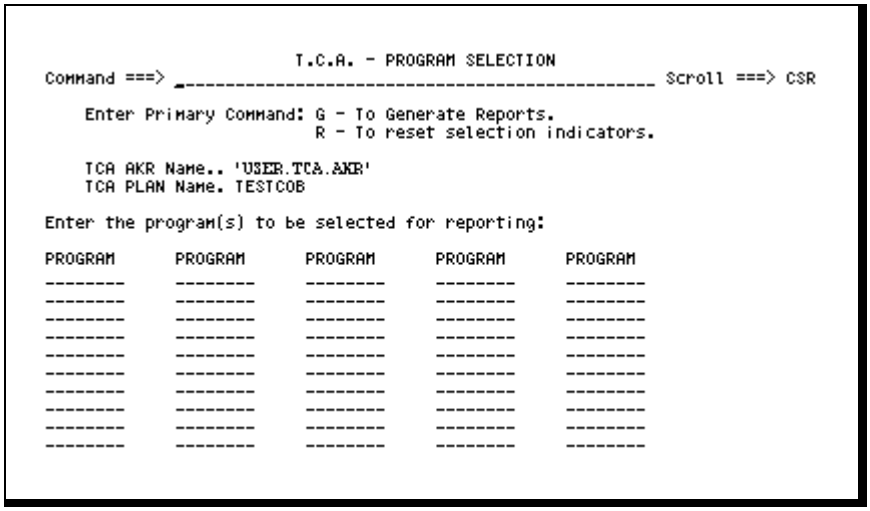
Selecting Specific Programs for Reports

To select specific programs for reports

1. From the T.C.A. - Result Selection screen (Figure 30 on page 55), select the test results for the programs on which you generate reports. You may select more than one test result.

2 Type P on the command line and press Enter. The T.C.A. - Program Selection screen, shown in Figure 31, displays.

Figure 31 • Program Selection

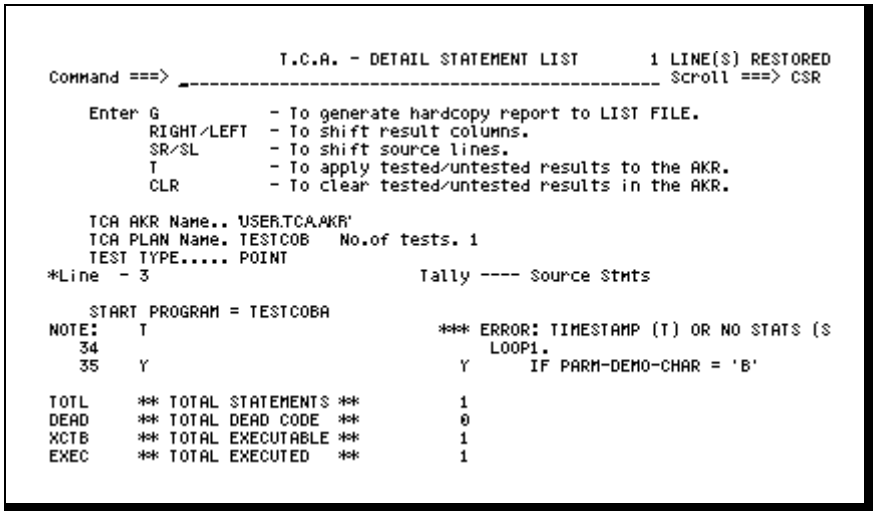


```
                        T.C.A. - PROGRAM SELECTION
Command ===> _____________________________________________ Scroll ===> CSR

    Enter Primary Command: G - To Generate Reports.
                           R - To reset selection indicators.

    TCA AKR Name.. 'USER.TCA.AKR'
    TCA PLAN Name. TESTCOB

Enter the program(s) to be selected for reporting:

PROGRAM      PROGRAM      PROGRAM      PROGRAM      PROGRAM
--------     --------     --------     --------     --------
--------     --------     --------     --------     --------
--------     --------     --------     --------     --------
--------     --------     --------     --------     --------
--------     --------     --------     --------     --------
--------     --------     --------     --------     --------
--------     --------     --------     --------     --------
--------     --------     --------     --------     --------
--------     --------     --------     --------     --------
--------     --------     --------     --------     --------
```

3 Type the program names, then type G on the command line and press Enter.

A screen showing the selected report displays. For example, if you selected Detail Count and Program Summary Report from the T.C.A. Report Selection screen, you would see a report similar to the one shown in Figure 32.

Figure 32 • Detail Statement List



```
                    T.C.A. - DETAIL STATEMENT LIST       1 LINE(S) RESTORED
Command ===> _____________________________________________ Scroll ===> CSR

    Enter G           - To generate hardcopy report to LIST FILE.
          RIGHT/LEFT  - To shift result columns.
          SR/SL       - To shift source lines.
          T           - To apply tested/untested results to the AKR.
          CLR         - To clear tested/untested results in the AKR.

    TCA AKR Name.. USER.TCA.AKR'
    TCA PLAN Name. TESTCOB   No.of tests. 1
    TEST TYPE..... POINT
*Line  - 3                          Tally ---- Source Stmts

    START PROGRAM = TESTCOBA
NOTE:    T                            *** ERROR: TIMESTAMP (T) OR NO STATS (S
   34                                       LOOP1.
   35    Y                              Y      IF PARM-DEMO-CHAR = 'B'

TOTL     ** TOTAL STATEMENTS **          1
DEAD     ** TOTAL DEAD CODE  **          0
XCTB     ** TOTAL EXECUTABLE **          1
EXEC     ** TOTAL EXECUTED   **          1
```

Generating Hardcopy Reports

To generate and print a report

1. From theT.C.A. - Result Selection, type `G` on the command line and press Enter. You receive a message confirming the length of the printed report listing.

2. Type `PLOG` to display the Options - Log/List/Punch Definition pop-up and press Enter. Note the message stating the list file is allocated.

3. Check that the job statement is correct.

4. Check that the List column Process option is appropriate.

5. Type `2` on the command line and press Enter to execute the job that processes the listing. When processing ends, you see the message `LIST OPTIONS PROCESSED`.

6. In ISPF, use the job output manager to print or view the report file.

Note:

In the report, each program starts on a new page. A program index is provided on the last page.

Applying Tested and Untested Results to the AKR

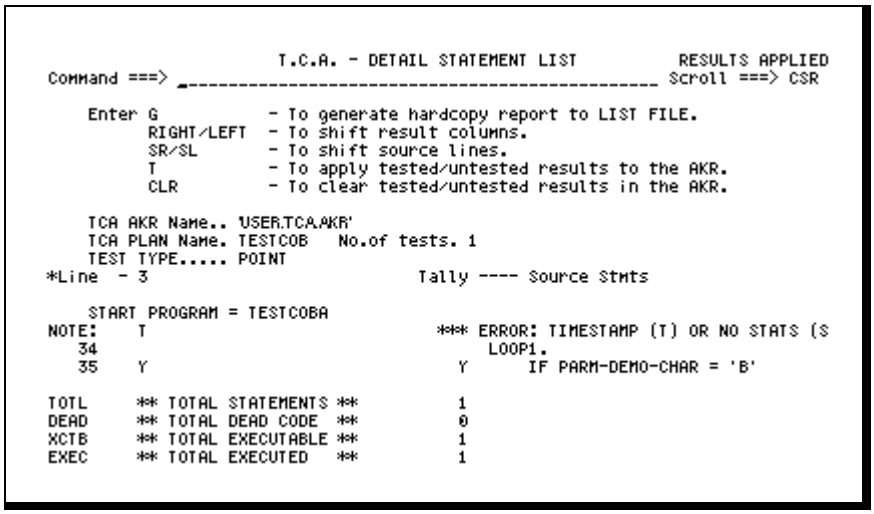
To identify the lines of tested and untested code

1 From the TCA - Detail Statement List screen, type `T` and press Enter to apply tested results to the AKR. The message `RESULTS APPLIED` displays in the upper right corner of the T.C.A. - Detail Statement List screen, as shown in Figure 33.

Note:

Use the CLR command to clear any tested or untested results in the AKR.

Figure 33 • TCA Detail Statement List

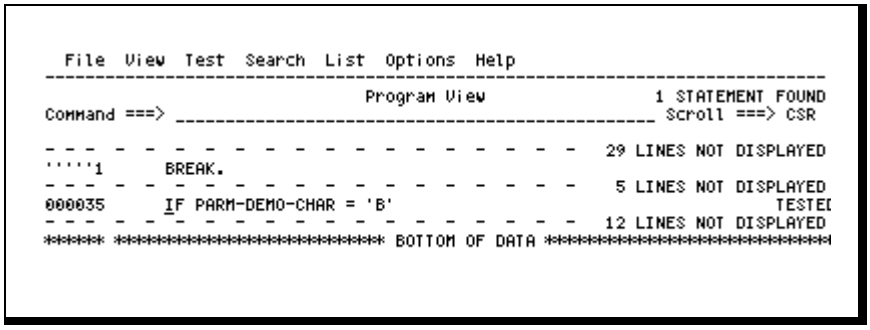


```
                        T.C.A. - DETAIL STATEMENT LIST          RESULTS APPLIED
Command ===> ____________________________________________________ Scroll ===> CSR

    Enter G           - To generate hardcopy report to LIST FILE.
          RIGHT/LEFT  - To shift result columns.
          SR/SL       - To shift source lines.
          T           - To apply tested/untested results to the AKR.
          CLR         - To clear tested/untested results in the AKR.

    TCA AKR Name.. 'USER.TCA.AKR'
    TCA PLAN Name. TESTCOB   No.of tests. 1
    TEST TYPE..... POINT
*Line  - 3                              Tally ---- Source Stmts

    START PROGRAM = TESTCOBA
NOTE:    T                                *** ERROR: TIMESTAMP (T) OR NO STATS (S
   34                                          LOOP1.
   35    Y                                 Y       IF PARM-DEMO-CHAR = 'B'

TOTL     ** TOTAL STATEMENTS **            1
DEAD     ** TOTAL DEAD CODE  **            0
XCTB     ** TOTAL EXECUTABLE **            1
EXEC     ** TOTAL EXECUTED   **            1
```

2 Type `VIEW` on the command line and press Enter to display the Program View screen.

3 Type `FX TES` and press Enter to view tested lines of code, as shown in Figure 34.

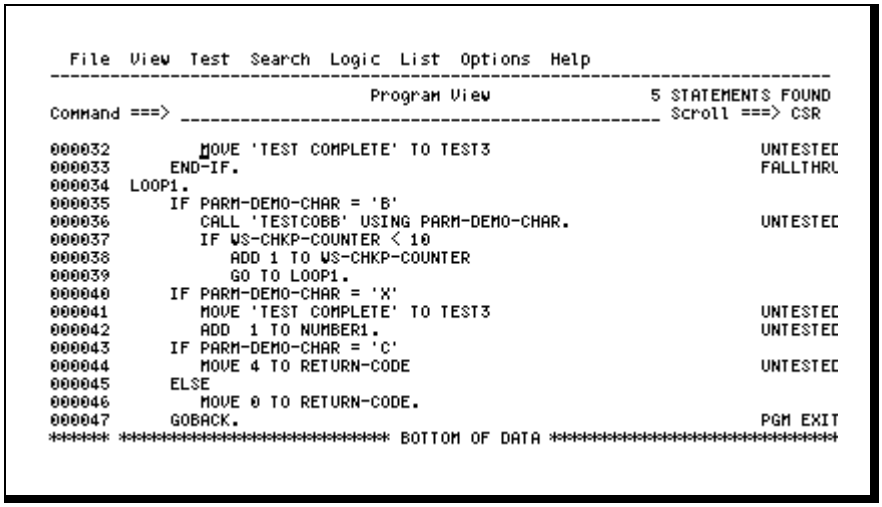
Figure 34 • Example of Program View Displaying Tested Code



```
  File  View  Test  Search  List  Options  Help
--------------------------------------------------------------------------------
                              Program View                1 STATEMENT FOUND
Command ===> ____________________________________________ Scroll ===> CSR

- - -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -   29 LINES NOT DISPLAYED
'''''1      BREAK.
- - -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -    5 LINES NOT DISPLAYED
000035      IF PARM-DEMO-CHAR = 'B'                                    TESTED
- - -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -   12 LINES NOT DISPLAYED
****** **************************** BOTTOM OF DATA ********************************
```

4 Type `FX UNT` and press Enter to view tested lines of code, as shown in Figure 35.

Figure 35 • Example of Program View Displaying Untested Code



11 TCA Commands, Scripts, and Profiles

This chapter describes the TCA screens and functions you can perform using the LIST TCA command as an alternative to the CUA interface features and contains these sections:

Note:

Detailed information about all SmartTest commands is provided in the *ASG-SmartTest Reference Guide.*

LIST TCA Command Parameters

Command	Description
LIST TCA	Displays the File - TCA Test Plan Selection pop-up, which is used to establish the TCA AKR and the TCA Plan name, and to select a TCA option.
Optional Parameters	
*	Displays the TCA - Active Plans pop-up listing all TCA Plans contained in the AKR. Use this pop-up to select or delete a plan for test coverage.

Command	Description
{*planname*}	Displays the TCA - Coverage Plan screen for a specific (new or existing) TCA Plan. Use this pop-up to define a TCA Plan, initiate a test session, maintain coverage results, and generate reports.
{*planname*} LIST	Displays the TCA - Test Utility screen that lists test results recorded under the current TCA Plan. Use this screen to maintain coverage results or select a result for another test.
{*planname*} GEN	Displays the TCA - Report Selection pop-up, which is used to select report options before generating reports.
{*planname*} SETUP	Displays the test session setup screen for the environment (i.e., TSO, CICS) associated with your TCA Plan. Use this pop-up to run test sessions on programs and to modify program environment settings.

Using Commands and Profiles with Scripts

Using the SmartTest-TCA command, SET SCRIPT ON, you can record TCA commands and associated screen commands into a script file. When executed, the script repeats the commands used in the session in which it was recorded. Creating scripts requires that you understand how each command works and how to sequence commands. You must also know which commands apply to which screens.

Screen-dependent Commands

This table provides a list of TCA screens and their valid commands. These are the commands that are recorded into the script files when you perform TCA testing with scripting on. You can also use these commands to build your own scripts.

Screen or Pop-up	Fields/Parameters	Field/Parameter Description
File - TCA Test Plan Selection	TAKR=*AKR name*	TCA AKR Name
	PLAN=*plan name*	TCA Plan Name
	S*n* where *n*=1 through 5	TCA Options

Screen or Pop-up	Fields/Parameters	Field/Parameter Description
T.C.A. - Coverage Plan	S	Set up a test session
	G	Generate reports
	L	List coverage results
	P*n*=*module.program* where *n*=program number	Add or modify corresponding module and program
TSO Session Setup	R	Run the test session
	PARM=*parameter*	Execution parameters
	CLIST=dataset name	Dataset name
	MEMBER=*CLIST member name*	Member
Batch Session Setup	S	Submit execution JCL
	SG	Submit generated JCL
	LOAD=*load module*	Load module
	PROC=*procedure name*	Procedure Name
	STEP=*step name*	Step Name
	BDSN=*batch JCL dsn*	Batch execution JCL: Dataset name
	BMEM=*batch JCL member*	Batch execution JCL: Member
	GDSN=*generated JCL dsn*	Generated JCL: Dataset name
	GMEM=*generated JCL member*	Generated JCL: Member
File - TCA Test Plan Run	DESC=*description*	Test Description
	S*n* where *n*=1 through 3	TCA Test Options

Screen or Pop-up	Fields/Parameters	Field/Parameter Description
T.C.A. - Report Selection	S*n* where *n*=1 through 5	TCA Report Options
T.C.A. - Result Selection	S*n* where *n*=result number	Selects result
	D*n* where *n*=result number	Deselects result
	R	Resets all selections
	A	Selects all results
	G	Generates report
	P	Selects programs for the report
T.C.A - Program Selection	G	Generates reports
	R	Resets all selections
	A	Selects all programs
	S*n* where *n*=program number	Selects a program
	D*n* where *n*=program number	Deselects a program
TCA Record	TAKR=*AKR name*	TCA AKR Name
	PLAN=*plan name*	TCA Plan Name
	DESC=*description*	Test description
TCA Report	G	Generates hardcopy report

Super Commands

Super commands are command phrases with parameters that allow you to perform some of the same functions represented by standard TCA commands using fewer lines of text. This table provides a list of TCA super commands and their parameters. Use these commands manually to expedite testing within TCA or within scripts. You can enter a single super command on multiple lines, since continuation occurs when a command line ends in a comma. However, parameters cannot span lines.

Command	Description
TCA DEFINE	Specifies an existing AKR to be used as the TCA AKR.
Parameters	TAKR=*AKR name*. You must define the TCA AKR before you can use this command to define a TCA plan.
TCA DEFINE	Defines a plan in a defined TCA-specific AKR.
Parameters	PLAN=*plan name*. ENV=*environment*. The valid values are 1 through 12. DESC=*description*. If you do not specify, the default is used. P*n* is the load module CSECT name, where *n*=1 through 50.
TCA RUN	Executes a test session using the information stored in the plan.
Parameters	PLAN=p*lan name*. TYPE=*test result type*. The valid values are COVERAGE and POINT. COVERAGE is the default value. DESC=*description*. If you do not specify, a default is used.
TCA LIST	Exports or deletes results in a plan.
Parameters	PLAN=*plan name*. EXPORT=*export result number*. DELETE=*result number*.
TCA RECORD	Records execution coverage information into a TCA Plan.
Parameters	TAKR=*AKR name*. PLAN=*plan name*. DESC=*description*.

Command	Description
TCA REPORT	Generates reports of results in a plan.
Parameters	PLAN=*plan name*. TYPE=*type of report*. The valid values are 1 through 5 and the default value is 1. RESULT=*test result number*. The valid values are 1 through x, or ALL. ALL is the default value. PROG=*program name*. ALL is the default value. HARDCPY=Y|N. Indicates whether a hardcopy report is produced. APPLY. Applies the tested and untested results to the AKR.

Captured Scripts Versus Super Scripts

The commands in these example scripts were saved using the `SET SCRIPT ON` command while the user executed a TCA test session.

Example Script Files

This script runs a single TCA test session:

```
LIST TCA              (Initial selection of TCA )
TAKR='MYTCA.AKR'       (Define TCA AKR )
PLAN=DEMO1            (Select DEMO1 plan )
S2                    (Selection display plan)
S                     (Select setup TCA test from Plan)
R                     (Run TCA test)
DESC=TEST OF SAMPLE   (Response to Prompt of Test)
S1                    (Test type selection and execution)
END
```

This script generates reports after completing several tests:

```
LIST TCA             (Initial selection of TCA )
TAKR='MYTCA.AKR'     (Define TCA AKR )
PLAN=DEMO1           (Select DEMO1 plan )
S2                   (Selection display plan)
G                    (Select generate TCA reports)
S1                   (Select Detail Count TCA report )
A                    (Select ALL results for report)
G                    (Generate report)
G                    (Generate Hardcopy )
PLOG                 (Select Deallocation)
3                    (Select VIALST and print report)
END
```

Both above scripts could be combined and run using super commands, as shown in this script:

```
TCA DEFINE TAKR='MYTCA.AKR'   (Define TCA AKR)
TCA RUN PLAN=DEMO1,TYPE=COVERAGE,DESC=TEST1 (EXEC test)
TCA REPORT PLAN=DEMO1,TYPE=1,RESULT=ALL,HARDCPY(Generate report)
PLOG (Select Deallocation)
3    (Select and print)
END  (End processing)
```

TCA Profiles and Scripts

To use the LIST PROFILES command in a script, follow this step:

- Type `LIST PROFILES PROF={`*dsn*`} S{`*n*`}`

  where:

  dsn is the name of the profile dataset.

  S*n* is a sequential number representing the profile.

  The valid values for *n* are 1 through 99.

Other Commands

Command	Description
ENV	Displays the Environment Selection pop-up (see Figure 11 on page 17). Use this screen to establish the environment, any settings it requires, the AKR where programs reside, and the load libraries to use during test sessions.
SET SCRIPT ON	Initiates scripting. Use this command to record scripts used to automate testing.

Index